XLitePro V2.0 - Online Help manual

1. About This Manual
2. Introducing to XLitePro
   - What is in XLitePro
3. The XLitePro Requirements
   - PC Hardware & Software Requirements
   - Host Requirements
4. Installing XLitePro
   - Running Setup
     - Extracting Package Installation Files
     - Running Setup
     - The Silent Installation Mode
     - Exporting XServer's Settings During Installation
   - Running UNINSTALL
   - Upgrading XLitePro
   - Multi-user Installation
5. The XLitePro Database
   - Keyboard Definition Files
   - The Color Definitions File
6. Configuring XLitePro
   - Using ComSetup
     - The Communication Setup Tab
   - The All Trace Tab
     - The Network trace Box
   - The XwpPeg Utility
     - The TCP/IP Retransmission Timeout Parameters
   - Using XSettings
     - The Startup Tab
       - The Window Mode List
       - The Use XDMCP Box
     - The Mouse_Keyboard Tab
       - The Mouse Settings Box
       - The Keyboard Settings Box
     - The Screen Tab
       - The Screen Settings Box
       - The GLX Extension Box
     - The Font Control Tab
     - The Advanced Tab
       - The Access Control Box
       - Some Hints about Authorization and Host Access Policy
     - The Troubleshooting Tab
     - The Trace Tab
       - The Network trace Box
   - The Run Box
     - The XServer Tab
7. Telnet
   - Starting and Terminating Telnet
     - Starting a Telnet Session
     - Terminating a Session
   - Details of a Session

Telnet Menu Options
The Help Menu
The Commands Menu
The Edit Menu
The View Menu
The Options Menu
The Settings Option
The Keys Tab
The Type Tab
The Text Tab
The Text Tab for the XTERM Type
The User Defined Tab
The Logging Tab
The ExtScreen Tab
The Keyboard Mapping Option
List Assigned Functions
Edit function
Terminal Emulation in Telnet
Running Telnet with Command Line Parameters
The "[TELNET]" Section of the ini-file
8. Startup
Starting and Terminating Startup
Running a Startup File
Executing a Command/File
Entering Startup Info
The Start Method Box
The Login Box
The Command Box
Startup Jobs
Saving a Job
Opening a Job
Running Startup with Command Line Parameters
The "[Startup]" Section of the ini-file
9. Using XServer
Starting XServer
Initiating Remote Login Sessions
The XDMCP Method
The Telnet Method
REXEC, RSH and RLOGIN Methods
Terminating XServer
A Note on Copy and Paste
Single Window Mode
Example of Initiating a Login Session by Using REXEC to Start xterm
Full Screen Mode
Example of Initiating a Login Session by Using XDMCP
Multiple Window Mode
Example of Initiating a Login Session by Using Telnet to Start xterm
The X Client's Menu
Multiple+RemoteWM Mode
XServer Commands
The Edit Menu
The Options Menu
The Macros Menu
The Run Menu
Other Commands on the X Client's Menu
Running XServer with Command Line Parameters
Running Several X-sessions
Using LBX
About LBX

About lbxproxy
Running xterm from Telnet
Running xterm from Startup
About xfindproxy
About proxymngr
Proxy Manager Config-File
Proxy Manager Details
The "[XSETUP]" Section of the ini-file
10. Font Control
Font Path
Font Directory
Font Server
Editing Font Path
Font Service
Understanding the Font List
Pseudo Fonts
Pseudo Fonts Directory
Creating Pseudo Font
Viewing Pseudo Font
Deleting Pseudo Font
Compiling Pseudo Font to the X11 format
Creating Pseudo Fonts on X Client Request
Enabling to Use
Restriction of Windows Fonts Accessibility
Appendix A Keyboard Mapping File Format
Keyboard Mapping File Format
The KEYS Section
The COMPOSERS_XKK Section
Available Keyboard Mapping Files
Appendix B Description of Terminal Capabilities
Description of Terminal Capabilities
Terminal Parameter Settings
Appendix C Troubleshooting
XServer & XSettings & Font Service
Appendix F Local X and Motif Clients
Local X and Motif Clients

1. About This Manual

This User's Manual describes how to install, configure and use the XLitePro package on a 32-bit IBM PC or compatible personal computer running one of the following operating systems: MS Windows 9x/ME/NT4/2K/2K3/XP/Vista. A small volume of the manual reflects simplicity of using this software tool created nevertheless on the basis of up-to-date information technologies.

The following items will be covered:

- Purpose and composition of XLitePro
- Hardware & Software requirements of XLitePro
- Installation procedure for XLitePro
- XLitePro database composition
- Configuring XLitePro
- XLitePro working sessions.

2. Introducing to XLitePro

XLitePro is a non-complicated product for integrating the Microsoft Windows and TCP/IP network environments. XLitePro is an inexpensive but effective way to transform a standard PC running under MS Windows (9x/ME/NT4/2K/2K3/XP/Vista) into a multi-function terminal. Being based on the TCP/IP open standards, the package integrates a PC into an interoperable computer network. The network of dissimilar computers and operating systems becomes perfectly transparent to you. XLitePro enables you to work on your PC's screen with several applications executed simultaneously on various network nodes. As a result, a heterogeneous network appears to you as a unified large computer system arranged directly on your desktop.

By using the industry standard Secure Shell (SSH1/SSH2) protocol for remote logins, intended to provide secure encrypted communications between two untrusted hosts over an insecure network, the package brings you typical remote system administrating, file transferring, and access to corporate resources over the Internet. With its SSH1/SSH2 features support, the package brings Security to your PCs, company LAN/WAN, or Intranet.

What is in XLitePro

XLitePro is an integrated and powerful 32-bit software tool consisting of the following functional parts:

Telnet virtual terminal emulator

Telnet is a communications and terminal emulation program for logging into remote machine and executing commands in a remote machine. It allows you to connect to and communicate with hosts that support the Telnet protocol and run a Telnet service over an insecure channel.

While you are using Telnet, you can:

- Initiate and control remote login sessions in the Telnet mode
- Set some options for particular implementations of Telnet
- Change fonts of text displayed in the Telnet window
- Select a terminal emulation mode in the Telnet session.

The Telnet program can emulate XTERM, AT386, ANSI, VT52, VT100, VT125, VT220 and VT240 terminals for character-mode applications. Advanced users can edit the terminal capabilities description file to suit to the special environment.

By using the **Keyboard Mapping** option (i.e. keymap editor invoking), you can load, change (re-define keys and create a new keyboard layout), and save any keyboard definition file.

The Startup Program

Startup is a program for automating host access with using the REXEC, RSH, RLOGIN, or Telnet protocol. REXEC and Telnet operations may be done either by direct access to a remote host or through one of the established SSH1/SSH2 protocol connections as well.

By using the Startup program, you can:

- Enter one or more commands in a single line and execute them on a host
- Run a local startup file (with a sequence of executable commands) on a host
- Enter a command line to run a local executable file (e.g., telnet, local X clients, etc) and execute it on your PC
- Create/save/select/remove/open/execute startup jobs.

Startup job is a task with a certain set of parameters (start method, login information, command

line, settings). You can create a job (i.e. store current parameters under a certain name). A job may be launched by clicking on its icon.

XServer

The XLitePro's XServer is a program that emulates the X terminal on your PC. It is the X-server implementation of the X11R6.6 release of the X Window System. XServer allows you to run one or more X Window based client applications (X clients) that are resident on host computers. The host can be any computer that supports the X protocol across your TCP/IP network.

X client applications can be displayed in individual windows, as multiple windows contained in a single XServer's window, or in a full-screen mode outside the Microsoft Windows graphical environment. The first two methods include functions to copy and paste data between X clients and the Microsoft Windows clipboard.

While running XServer, you can initiate a login session by using Telnet, XDMCP, or Startup (REXEC, RSH, RLOGIN) methods. You can launch several X-sessions that will be running simultaneously each with its own settings. XServer can work on multi-monitors PCs.

While running XServer, you can launch any local X client supplied with the package.

The macro options of XServer let you create macro-files to remember sequences of your keystrokes. Then, whenever you want, you can play back these keystroke macros in a target X-application window by choosing an appropriate macro-file.

XServer provides German and/or French messages and text labels under the corresponding national version of MS Windows.

XServer supports the following X Extensions:

- Extended-Visual-Information
- XinputExtension
- MIT-SUNDRY-NONSTANDARD
- SHAPE
- XTEST
- XC-MISC
- XC-APPGROUP
- GLX for OpenGL
- RENDER
- SECURITY
- LBX

XServer can connect to a remote LbxLoxy service (a Low Band Width X proxy) either by direct access to a remote host or through one of the established SSH1/SSH2 protocol connections as well.

XServer can use MESA (6.2.1), an open-source implementation of the OpenGL specification.

Font Service

Fonts installed on your system can be used by any graphical X Window System application. A remote font server is started automatically with your graphical environment, and then all fonts installed on your system are automatically available to all of your X Window System applications. X clients

request character information from the font server, which can return data in various formats, ensuring high quality display of a variety of font styles and sizes.

You can use your familiar host-based X fonts via XLitePro's XFontset Service. The XFontset Service utility provides:

- Rasterized fonts with outline data to all X applications
- Wide range of local X fonts (the standard font sets supplied by MIT along with X11 R6)
- Support for different locale 16-bits fonts including Chinese, Japanese, or Korean fonts, etc.
- Use of uncompressed PCF, SNF, and BDF format files as well as these fonts compressed with the 'compress' or 'gzip' utility
- Font Compiler to compile Microsoft Windows fonts into the X11 format
- Microsoft Windows fonts support in X-sessions.

Local X Clients

While running Startup or XServer, you can launch the following local X clients supplied with the package:

- **mwm** (based on OpenMotif 2.0)
- **glxinfo** (the GLX extension local X Client)
- **glxgears** (the GLX extension local X Client)
- **xclock**
- **xdpyinfo**
- **xev**
- **xfd**
- **xfontsel**
- **xkill**
- **xlsfonts**
- **xmodmap**
- **xprop**
- **xrdb**
- **xset**
- **xshowcmap**
- **xwininfo**

You can use the local X clients when remote ones are inaccessible or inconvenient to run. You can launch the local X clients the same way as remote ones using their command line syntax.

< **previous** | **content** | **next** >

3. The XLitePro Requirements

Your computer system must meet the following hardware, software, host and network requirements for you to install and use XLitePro.

PC Hardware & Software Requirements

- A standard 32-bit IBM PC (e.g., i486, or Pentium) or 100% compatible
- 8 Mbytes RAM
- Color graphics controller supporting SVGA video modes
- Mouse Unit compatible with Microsoft Windows
- 22 Mbytes free hard disk space.

Note that this disk requirement does not account for the disk cluster size. The larger the cluster size the greater the disk requirement.

In addition to the above requirements, you need one of the following operating systems:

- MS Windows 9x/ME/NT4/2K/2K3/XP/Vista
- TCP/IP facility with Windows Sockets Interface.

Host Requirements

- TCP/IP protocols over Ethernet or Serial port connection
- Virtual terminal protocol Telnet
- Login account on the host machine.

4. Installing XLitePro

This chapter describes how to install the XLitePro software. The chapter assumes that you have one of the Microsoft Windows 9x/ME/NT4/2K/2K3/XP/Vista operating system installed as described in the corresponding user's guide for the product.

This chapter and the rest of the manual refer to the following two installation directories whose names you should specify at the installation steps:

- the home (or destination) directory (in which you install the package components files for all users)
- the configuration files directory (in which you store your particular configuration files, e.g. ini-files).

If you install XLitePro in a directory different from the default, simply supply your directory name when appropriate directories are requested.

The installation of the package is carried out by running the Setup program.

Running Setup

The software product comes normally as a self-extracting archive file that contains the package installation files.

The installation procedure consists of two steps:

1. Extracting the package installation files into a temporary distributive directory

2. Running Setup from the distributive directory.

You can run Setup manually in the default mode or use the "silent" installation mode of Setup to simplify multiple secondary installations.

Extracting Package Installation Files

This section describes how to extract the package installation files from the self-extracting file (compressed and created by WinZip).

To extract the package installation files, you should do the following:

1. Download the self-extracting archive file to your hard disk

2. Execute the file and select **OK**.

In the dialog box that appears, you can enter a name for a temporary distributive directory in the **Unzip to folder** edit field or use the **Browse** button to select it. By default, the installation files will be extracted to the distributive directory shown in the field.

Later (after successful installation), you can remove the temporary distributive directory or use it to start Setup for multiple secondary installations.

3. Choose **Unzip** to start extracting the files and then installing the package automatically. The archive file will be uncompressed and the installation files will be placed in the specified distributive directory.

4. After extracting the package installation files, choose **OK** in the box appeared.

The installation procedure (i.e. running Setup) will start automatically from the temporary distributive directory if you enable the **When done unzipping open: .\setup.exe** check box.

Running Setup

As soon as you start the installation process, you will see a number of dialog boxes with instructions for each installation step. These boxes have three buttons. The **Cancel** button quits the installation process. The **Back** button returns you to the previous step. When you press the **Next** button, the Setup program proceeds to the next installation step.

At any step of installation, you can use the **Cancel** button. The **Exit Setup** window appears.

You can confirm exiting or choose to continue installing.

For the first installation of the package, the procedure steps are as follows (with the dialog's names):

- **Welcome**

At this step, it is strongly recommended that you exit all MS Windows programs before running Setup.

- **Software License Agreement**

At this step, you must choose whether you accept all the terms of the Software License Agreement shown within the window or not before running Setup.

- **User Information**

In this box, you should enter the Person name, the Company name, and the product serial number (for registration purposes).

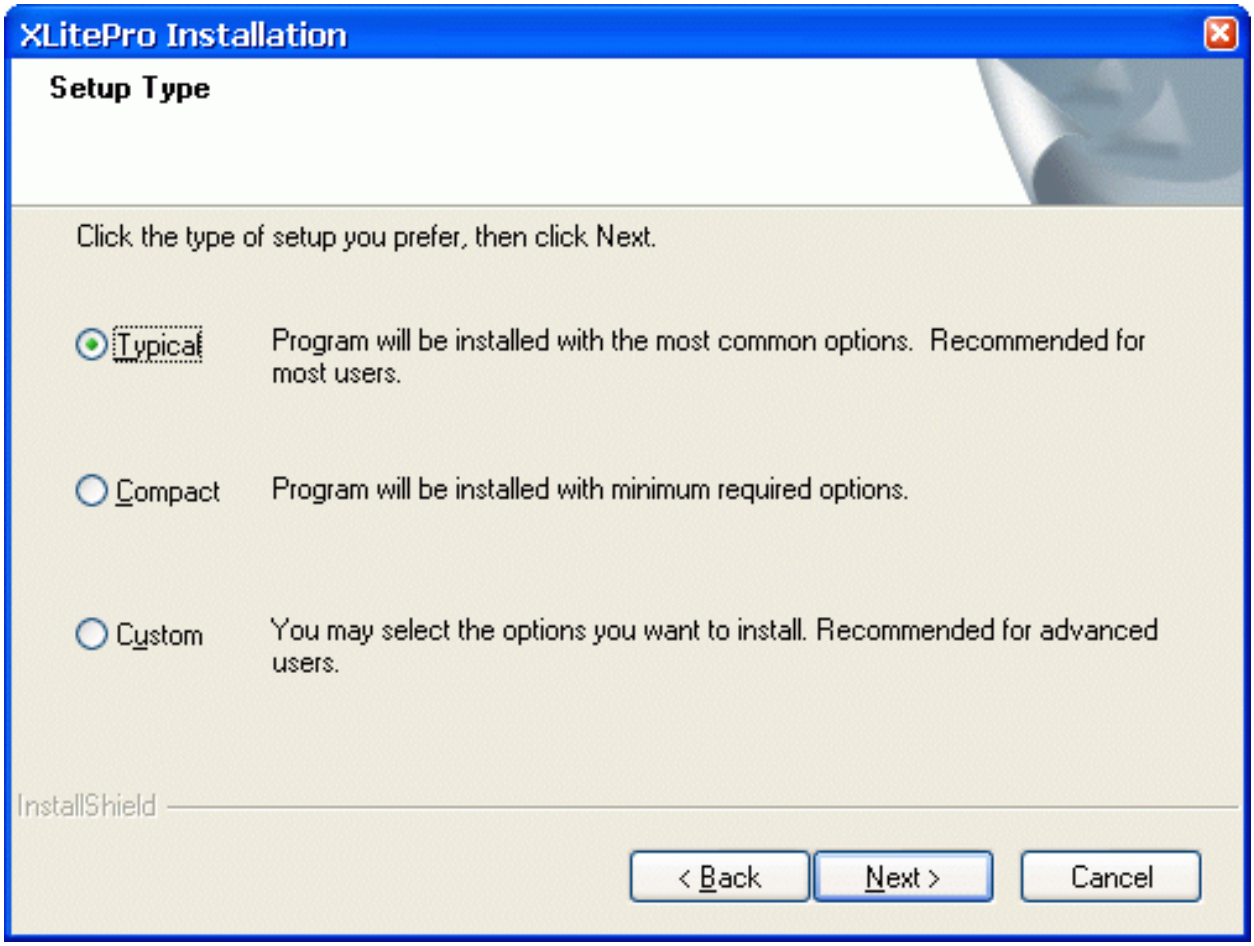
- **Choose Destination Location**

In this box, you should specify the folder where the package will be installed in (i.e. the home directory for the package). You can use the default folder name, enter your destination folder name, or select another folder using the **Browse** button.

To install to the specified folder, click **Next**. If the folder does not exist, Setup will create it. If Setup detects the package in the directory you specified, then it will prompt you to upgrade the package. (See section **Upgrading XLitePro** below.)

- **Setup Type**

At this step, you can choose components of the package you want Setup to install.

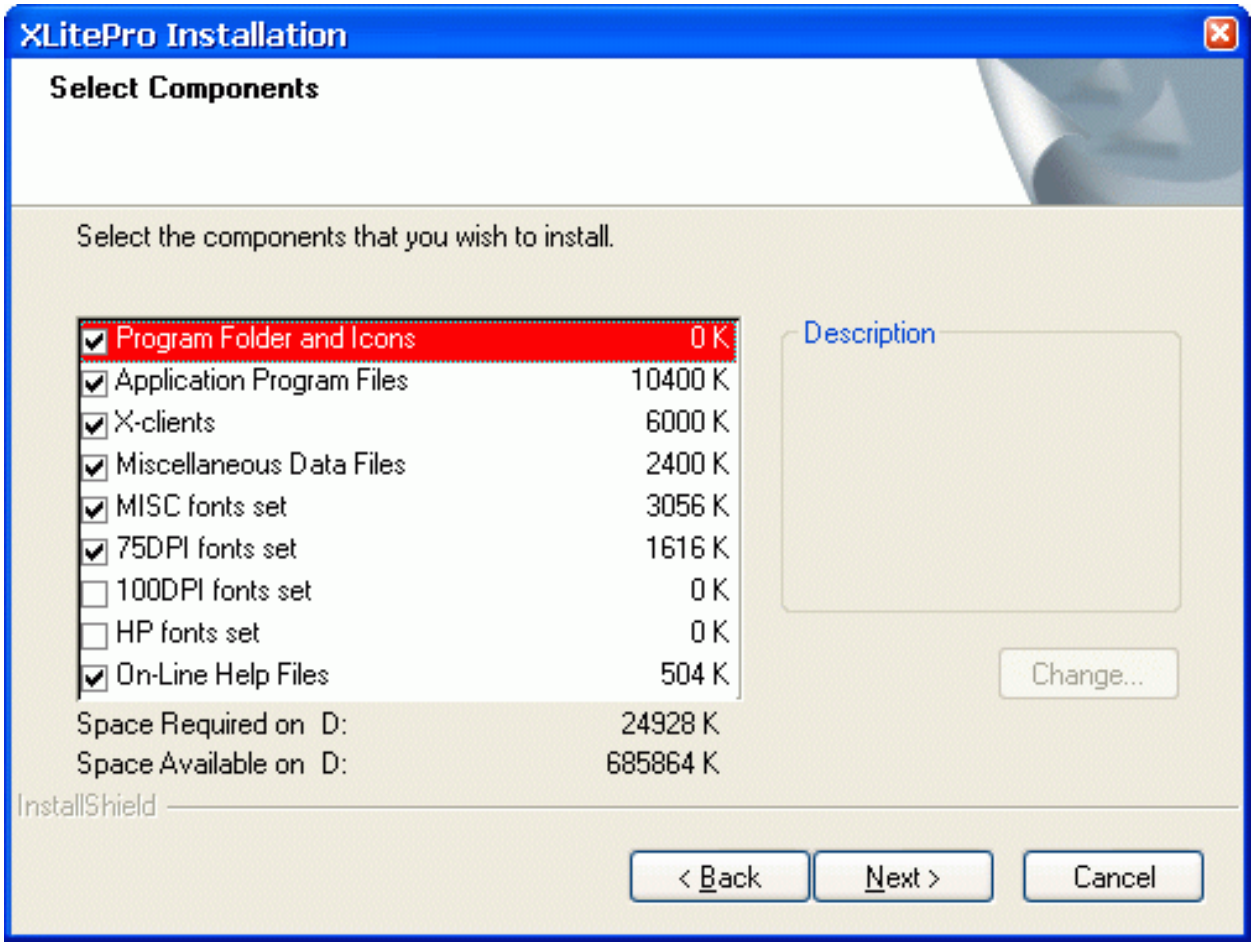


There are three Setup types: **Typical**, **Compact**, and **Custom**.

For the **Typical** type, Setup installs all the components of the package (the most common options).

For the **Compact** type, Setup installs the package without On-Line Help Files (the minimum required options).

For the **Custom** type, the **Select Components** dialog box appears on your display. Choose components that you wish to install (clicking on the checkmark positions).



At the bottom of the window, you can see how much **Space Required** will be used to install the components you choose and **Space Available** on your disk.

- **Select Program Folder**

At this step, you should specify the Program Folder to contain shortcuts for the components being installed. In the **Select Program Folder** dialog box, you can use the default folder name, enter a new folder name, or select another folder from within the **Existing Folders** list.

Setup will add program icons to the Program Folder specified.

- **Setup**

With the data specified, Setup begins to install the package. It shows how files of the components are being installed into the destination directory. Finally, Setup creates the XLitePro Program folder with shortcuts for the package components. It also adds the XLitePro item to the Programs menu.

- **Setup complete**

At this step, Setup informs you that the package is ready to run. In the **Setup complete** dialog box, click **Finish** to complete Setup.

You may run the installed programs by clicking program's icons from the XLitePro Programs' folder.

The Silent Installation Mode

The "silent" installation mode of Setup may be used for multiple secondary installations on different computers working with similar package installation environment (i.e. drive letters, installation directory name, presence of the package, etc).

The "silent" installation process consists of the following two steps:

1. Normal first installing of the package with creating a script file
2. Using the script file for multiple secondary installations.

1) To install the package with creating the script file, **setup.iss**, run Setup with the options:

setup -r -f1PATH\setup.iss

The **-f1** option enables you to specify an alternative file location and script file name. It is recommended to specify the absolute PATH for the option.

With these arguments, Setup performs normal installation of the package and creates the script file you specified in the command in the directory according to the specified PATH. The file contains data you specified for Setup to install the package. (See section **Running Setup** above.)

Setup performs secondary installations using the script file, so you have to specify no input data (you only watch "silently" how Setup works automatically).

2) To perform secondary installation, make sure that the script file created at the first step is located in the distributive directory (where the **setup.exe** file exists). If not, then copy the script file to the distributive directory, and then run Setup with the following command line:

setup -s -wauto

With this argument (and without the **-f1** option), Setup will install the package according to the **setup.iss** script file. Setup will search for the file in the distributive directory.

Also, you can use the following command line to perform secondary installation:

setup -s

This is the same as the "-wauto" option with the only difference: the Finish dialog message is suppressed in this case.

When running an InstallScript MSI or InstallScript installation in silent mode (i.e., using the **-s** option), the log file, **setup.log**, is by default created in the same directory and with the same name (except for the extension) as the response file. The **-f2** option enables you to specify an alternative file location and log file name. It is recommended to specify the absolute path for the option as in the following example:

.\PackInstall\Setup.exe -s -f1C:\PackInstall\Setup.iss -f2C:\PackInstall\Setup.log

After Setup has finished (successfully or not) you can find the ASCII tracing file, **mkltrace.out**, in the distributive directory and look it through for error messages. Note that the "silent" installation may require interactive actions if Setup detects serious problems.

Exporting XServer's Settings During Installation

You can port XServer's settings on other PCs by creating and using the **xsetup.ini** file as follows:

1. Install the package and save its distributive directory
2. Make desired XServer's settings (resulting in the "[XSETUP]" section in the **xwp.ini** file under the home directory)
3. Copy the "[XSETUP]" section from the **xwp.ini** file into the **xsetup.ini** file except line "FontPath"
4. Put the **xsetup.ini** file into the distributive directory of the package
5. Use your new distributive directory (with the **xsetup.ini** file) as a distributive directory for the secondary "silent" installations on other PCs. The resulting home directory of the package will contain the **xwp.ini** file with your "[XSETUP]" section in it.

Telnet

The Installation procedure (Setup) can read in a file you prepared beforehand to contain your settings for Telnet. The file must have the **prosets.ini** name and locate in the package distributive directory (where the **setup.exe** file is located). The settings will be placed in the **xwp.ini** file. This is especially useful for "silent" installations.

< previous | content | next >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

Running UNINSTALL

You can uninstall the package by choosing the **Uninstall** item from the Program Folder. The program will prompt you to confirm removing the package from your computer.

When **Uninstall** completed, some elements might not be removed. You should manually remove items related to the application.

Upgrading XLitePro

If you specify the home directory that contains the package installation files, the Setup program will detect it and prompt you to upgrade or configure it.

If you choose **YES**, i.e. to upgrade the installed package, Setup brings up the **Upgrade Type** window. You can choose **Typical**, **Compact**, or **Custom** installation type (like you do it at step **Setup Type** of the normal installation sequence). This allows you to upgrade the package or to reconfigure it without reinstalling binary programs.

If you choose **NO**, Setup returns you to the **Software License Agreement** step of the normal installation sequence. This allows you to completely reinstall the package.

If you specify the destination directory that exists, Setup brings up the **Choose Installation Type** window.

In this box, if you choose **Reinstall**, Setup will return you to the **Setup Type** step.

If you choose **Upgrade/configure**, Setup returns you to the **Upgrade Type** step. If you choose **Only Configure**, Setup goes to the **Select Components** step. Then, in both cases, the **Choose Configuration Location** window appears.

In this box, you should specify the configuration directory (i.e. configuration path) in which you store your particular configuration files (e.g. ini-files) and the configuration components you selected. Then, Setup prompts you to **Select Program Folder**.

Note that if you choose **Only Configure,** then Setup will make no changes in the home directory of XLitePro detected.

Multi-user Installation

Multi-user installation is intended for installing a single copy of XLitePro on a disk that will be shared by multiple users. XLitePro must be configured for each user on each PC it will be used on. Corresponding files that define user's local XLitePro configuration will be created in a specified directory (see **Choose Configuration Location**).

Multi-user installation is carried out as follows:

1. By running the Setup program, install the package on a disk that will be used for storing the shared copy (i.e. under the home directory of XLitePro).

The next step must be done by every user of the shared copy.

2. Run the Setup program. In the installation dialogs, you must specify the home directory of XLitePro for the shared copy (step **Choose Destination Location**), and a local directory where files defining a particular package configuration will be resident (step **Choose Configuration Location**).

Note that these secondary steps make no changes in the home directory of XLitePro created at the first step.

5. The XLitePro Database

The package database is represented by two ASCII files

- **terminfo.ini**
- **rgb.txt**

and also by 35 keyboard definition files with the file name extension **.KMF**.

The **terminfo.ini** file contains information for terminal emulation and is described in Appendix B.

Keyboard Definition Files

XLitePro has 35 keyboard definition files allowing you to use one of the 35 international PC keyboards. Each of them corresponds to the country your keyboard was designed for. These files are listed in Appendix A.

Your Keyboard Definition file has the **.KMF** extension. It resides in the XLitePro's configuration files directory (in the location you specified when installing XLitePro).

The basic purpose of a keyboard file is to assign PC keys to generate specific keysyms. A keysym is a key code that corresponds to a specific symbol supported by the X protocol.

A Keyboard Definition file is an ASCII source file that defines what key sequence is sent to a client when you press a given key on your PC's keyboard (i.e. keyboard mapping).

You can customize a keyboard by one of two ways:

- By modifying a selected (on installation) keyboard file
- By choosing the **userkbd.kmf** keyboard file and then editing it.

These are some of the things you can do:

- Make any key on your keyboard send any supported X keysym to the host
- Make use of extra keys on non-standard keyboards to send special keysyms to the host or to a client.

You can view and modify Keyboard Definition Files by using the Telnet's **Keyboard Mapping** option. (See section **The Keyboard Mapping Option** in Chapter **Telnet**).

The Keyboard Mapping File format is described in Appendix A.

All keyboard files are written for keyboards with a separate cursor keypad. Note that there are two U. K. keyboard files supplied. One is for a 101-key U.K. keyboard, and other is for a 102-key keyboard. The 101-key U.K. keyboard is identical to the U.S. keyboard except that holding **Shift** and pressing **3** produces a **POUND** sign instead of a '**#**' sign.

To input the **Euro** currency sign, the recommended **Alt_R+E** combination was inserted into the following KMF-files:

us15.kmf, **danish.kmf**, **belgian.kmf**, **decemfrc.kmf**, **decemfr.kmf**, **decemgr.kmf**, **decemuk.kmf**, **dutch.kmf**, **frencan.kmf**, **french.kmf**, **german.kmf**, **hungarn.kmf**, **italian.kmf**, **latinam.kmf**, **norwegia.kmf**, **portugue.kmf**, **slovenia.kmf**, **spanish.kmf**, **swedfinn.kmf**, **swedish.kmf**, **swissfre.kmf**, **swissger.kmf**, **uk102.kmf**, and **uk102m.kmf**.

The **msus.kmf**, **uk101.kmf**, **decemus.kmf**, **userkbd.kmf**, **us.kmf**, **dvorak.kmf**, and **jpn106.kmf** files do not provide for the **Alt_R+E** input.

The **us15.kmf** file is a copy of the **us.kmf** file with the **Alt_R** key description changed from **XK_Alt_R** to **XK_Mode_switch**.

In many European languages (e.g., France, Germany), users need to enter some special characters by combining a Diacritic (or composer) character and a normal letter. KMF files allow for this feature for national keyboards.

Note: if you need to send the four MS Windows specific key combinations to X clients, you have to enter the substitution strings into the [XSETUP] section of the **xwp.ini** file (for package's communication programs can use them).

For example:

CtrlEsc2=255.173	To send **Ctrl + Esc**, press **Ctrl+KEYPAD MINUS SIGN**
CtrlAlt2=255.171	To send **Ctrl + Alt**, press **Ctrl+KEYPAD PLUS SIGN**
AltEsc2=255.173	To send **Alt + Esc**, press **Alt+KEYPAD MINUS SIGN**
AltTab2=255.171	To send **Alt + Tab**, press **Alt+KEYPAD PLUS SIGN**

< previous | content | next >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

The Color Definitions File

The **rgb.txt** file shipped by MIT jointly with the X Window System contains predefined colors in the form of correspondence between their specific RGB values and symbolic color names. The color values are present in the **RRR GGG BBB** format, where R, G and B represent single decimal digits and determine the intensity of the red, green and blue primaries that make up each color.

Some sample lines in the **rgb.txt** file may look like the following (in the "RRR GGG BBB color's alias" format):

```
160  32 240 purple
255   0   0 red
255   0 255 magenta
```

6. Configuring XLitePro

This chapter describes how to configure the XLitePro package with the configuration utilities. The utilities allow you to set up XLitePro for your preferences, your host system and your PC.

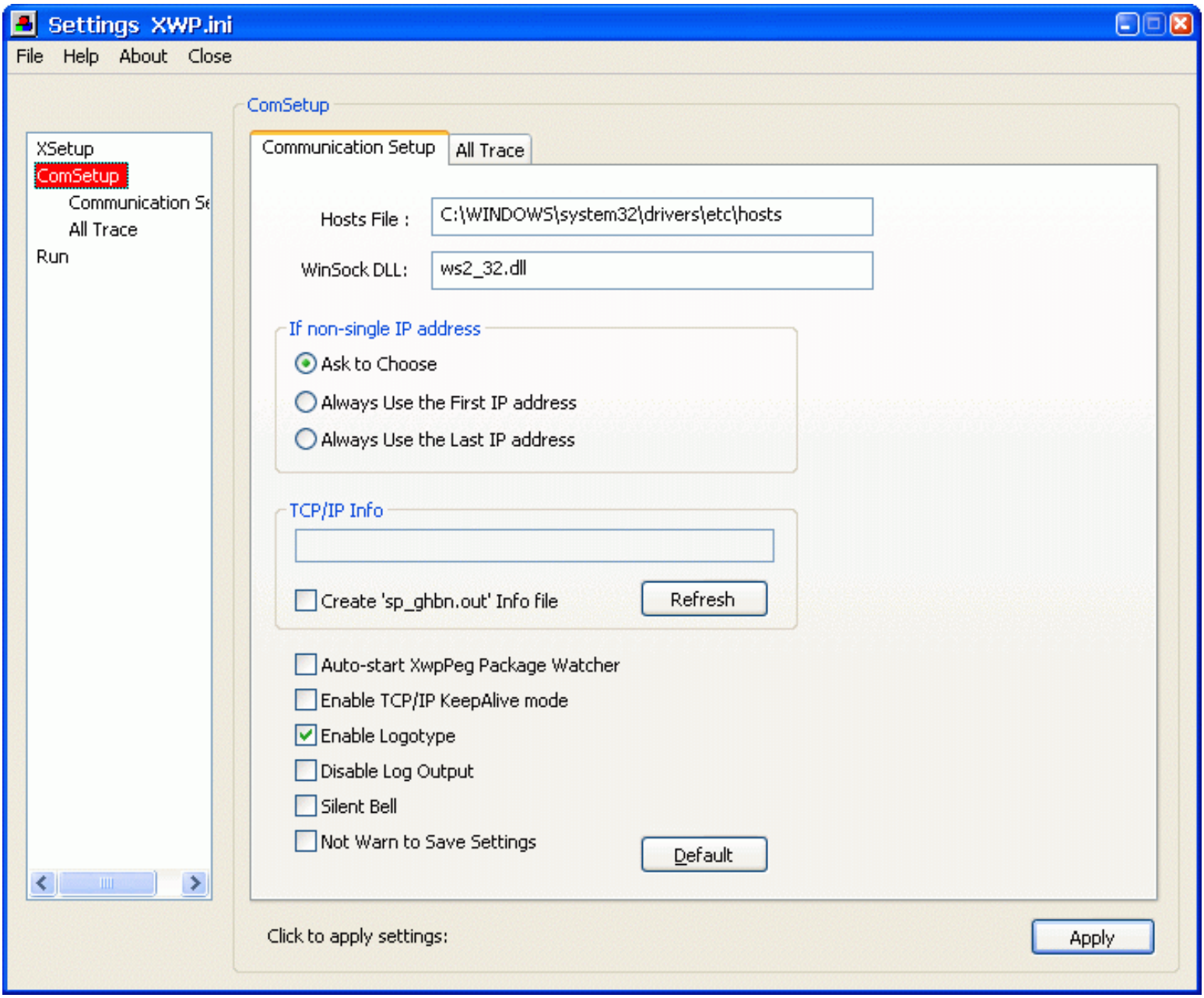
XLitePro supplies two configuration utilities:

- **ComSetup** allows you to make communication settings relating to the networking aspects of XLitePro that operate with the TCP/IP transport interface
- **XSettings** is used to make control settings relating to all aspects of XLitePro (mainly for XServer) except networking.

Using ComSetup

You can start ComSetup by clicking on the **Settings** item in the XLitePro Programs' folder (i.e., from the Start/ Programs/XLitePro menu).

The **Settings** window will appear on your display:



The **File / Open** menu item of the **Settings** window lets you choose an **.ini** file (**xwp.ini** by default), then load its settings and make them current for applications to run. You can edit the settings before running your applications.

The **File / Save as** menu item of the **Settings** window lets you choose an **.ini** file (**xwp.ini** by default) to save current settings.

The **Close** menu item of the **Settings** window simply closes it.

By clicking **Apply**, any new settings you make to the **Communication Setup** tab will be saved in the **xwp.ini** file (by default).

The following sections are available:

- The Communication Setup Tab
- The All Trace Tab
- The XwpPeg Utility

The Communication Setup Tab

The following input fields are available:

Hosts File

This field is used to specify a location of the **hosts** file. You must enter the **hosts** file that is used by MS Windows (**C:\WinNT\System32\drivers\etc\hosts** for MS Windows NT4/2000/XP and **C:\Windows\hosts** for MS Windows 9x/ME).

The **hosts** file is a list of remote computers in the standard format (IP-address hostname aliases). The contents of the **hosts** file will be used by other programs of the package when you want to select a host to connect to from the host list box.

WinSock DLL

This field specifies the DLL that provides Windows Sockets Interface to existing TCP/IP stack. By default, **wsock32.dll** of the Microsoft Windows' TCP/IP will be used. You can specify to use any other TCP/IP stack by entering its 32-bit Windows Sockets Interface DLL.

Auto-start XwpPeg Package Watcher

With this check box enabled, the XwpPeg utility is started automatically with starting any package's application. Otherwise, you can launch XwpPeg manually.

Enable TCP/IP KeepAlive mode

This check box specifies the package's applications to use the TCP/IP feature, KeepAlive, when communicating with remote computers over your network. When enabled, this prevents your connection from interrupting by a remote computer when your PC does not send messages to it for a long time.

Enable Logotype

This check box toggles displaying the Logotype image each time the package's application starts up.

Disable Log Output

When selected, this check box prevents any program of the package from writing log information to the ".out" and ".ini" files.

Silent Bell

When enabled, this check box will block all sounds from all applications of the package (i.e., internal sounds and the TCP/IP protocol bell requests).

Not Warn to Save Settings

Normally, you should not forget to **Apply** your changes to a tab when you leave it.

If this check box is clear, then the warning will appear each time you take to another tab or section of the window.

The warning will not appear if you select the check box.

Default

This button will initialise all these parameters to their default values.

The If non-single IP address Box

If your PC has more than one IP address (i.e. 'multi-home' PC – with non-single TCP/IP stack, e.g. for Ethernet + modem), then you should specify a mode for choosing one of them.

Always Use the First IP address
Always Use the Last IP address

These modes allow XLitePro's programs to automatically choose the local IP address.

Ask to Choose

You can set up this mode to specify that you will choose the address in the box brought up by the programs. The default mode is **Ask to Choose**.

The TCP/IP Info Box

Refresh

When you click on this button, ComSetup will search for available TCP/IP information and, if found, display in the info field the IP address and name of your PC according to mode settings.

Create 'sp_ghbn.out' Info file

If this check box is enabled, then all information found will be stored in the file. This allows you to check accessibility and obtain description of the TCP/IP stack used.

Using XSettings

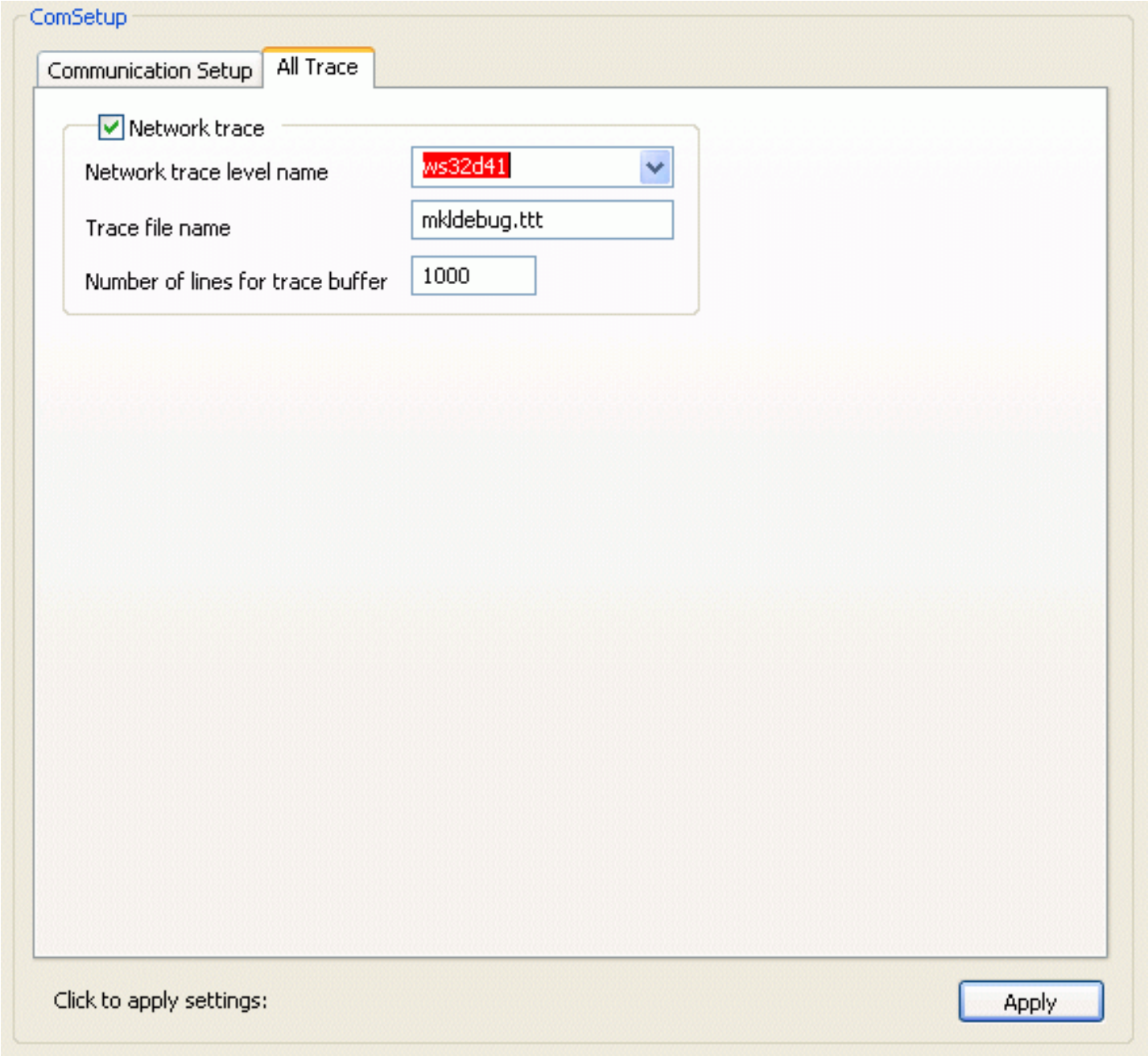
The **XSetup** dialog box consists of 7 tabs described below:

- The Startup Tab
- The Mouse_Keyboard Tab
- The Screen Tab
- The Font Control Tab
- The Advanced Tab
- The Troubleshooting Tab
- The Trace Tab

By clicking **Apply**, any new settings you make to the **XSetup** box will be saved in the **xwp.ini** file (by default).

You can cancel any changes you have made to the box and close it by pressing **Close**.

The All Trace Tab



Options that you can specify within this tab are for debug purposes. They exert influence on all applications of the package.

The Network trace Box

When the **Network trace** check box is selected, it enables network tracing for applications of the package.

Network trace level name

A value in this field that you can choose from the list box denotes a tracing level for collecting network trace information to the log file.

Trace file name

In this entry field, you can specify a name for a log file to store network trace data.

Number of lines for trace buffer

In this entry field, you can type in a length for the allocated memory (in number of lines) to flush it to the trace file.
The value of 0 means a "very large" buffer (as much as your system allows).
Caution: when a system crash happens, the data in the allocated memory is lost.

The XwpPeg Utility

XwpPeg is a program designed to check the state of the package license accounts. XwpPeg informs users about exceeding the allocated license accounts over the specified limit. If all users of the same package have XwpPeg running simultaneously with running any package's application (and terminate after completion of the last one) then all those users will have true information about the number of concurrent package users who use package license accounts (i.e. the allocated accounts).

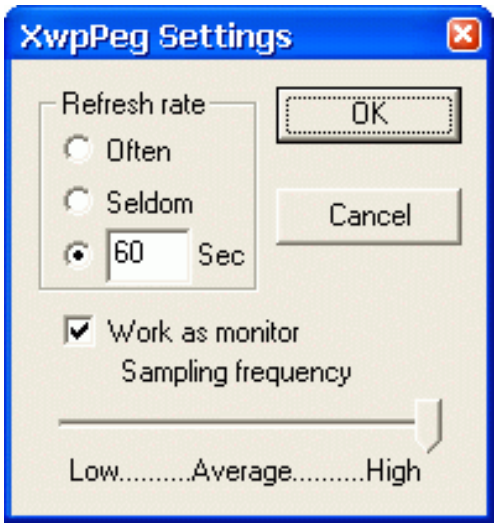
You can launch XwpPeg either automatically or manually. With the **Auto-start XwpPeg Package Watcher** check box enabled, XwpPeg is started automatically with starting any package's application.

Users can independently launch XwpPeg (before starting any package's application) to check/allocate one license account in advance. If the user starts XwpPeg manually then he/she must also manually terminate it to free the package license account allocated. Since XwpPeg is only informational, users can freely terminate it at any time; however they cannot automatically check the current number of concurrent users of the same package in that case.

You can start XwpPeg by double-clicking on the **XwpPeg** icon in the XLitePro Programs' folder:

If you do not want to display the XwpPeg window, you should set to zero the value of the WorkMonitor variable in the "[XWPWATCH]" section of your **xwp.ini** file (e.g., **WorkMonitor=0**). To display the window, set the value to unit.

If you click on the **Settings** button, the following dialog box appears:



When you have made desirable settings, press **OK**.

The TCP/IP Retransmission Timeout Parameters

TCP starts a retransmission timer when each outbound segment is handed down to IP. If no acknowledgment has been received for the data in a given segment before the timer expires, then the segment is retransmitted, up to the TcpMaxDataRetransmissions times. The default value for this parameter is 5.

When a TCP connection is established, the retransmission timer is initialised to three seconds; however, it is adjusted on the fly to match the characteristics of the connection using Smoothed Round Trip Time (SRTT) calculations (as described in RFC793). The timer for a given segment is doubled after each retransmission of that segment. Using this algorithm, TCP tunes itself to the normal delay of a connection. TCP connections over high-delay links will take much longer to time out than those over low-delay links.

By default, after the retransmission timer hits 240 seconds, it uses that value for retransmission of any segment that needs to be retransmitted. This can be a cause of long delays for a client to time out on a slow link.

MS Windows NT4/2000 provide a mechanism to control the initial retransmit time, and then the retransmit time is self-tuning. The following is based on the Microsoft Knowledge Base and Microsoft MSDN Library.

To change the initial retransmit timeout parameters, you can modify the following values in the following registry key:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters

Value Name: InitialRtt
Data Type: REG_DWORD
Valid Range: 0-65535 (decimal)
Default: 0xBB8 (3000 decimal)

The **InitialRtt** parameter controls the initial retransmission timeout used by TCP on each new connection. It applies to the connection request (SYN) and to the first data segment(s) sent on each connection. For example, the value data 5000 decimal sets the initial retransmit time to five seconds.

Value Name: TcpMaxDataRetransmissions
Data Type: REG_DWORD - Number
Valid Range: 0 - 0xFFFFFFFF
Default: 5

The **TcpMaxDataRetransmissions** parameter controls the number of times TCP retransmits an individual data segment (non-connect segment) before aborting the connection. The retransmission timeout is doubled with each successive retransmission on a connection. It is reset when responses resume. The base timeout value is dynamically determined by the measured round-trip time on the connection.

Caution

The above text contains information about editing the registry. Before you edit the registry, make sure you understand how to restore it if a problem occurs. Using Registry Editor incorrectly can cause serious problems that may require you to reinstall your operating system. Microsoft cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use

Registry Editor at your own risk.

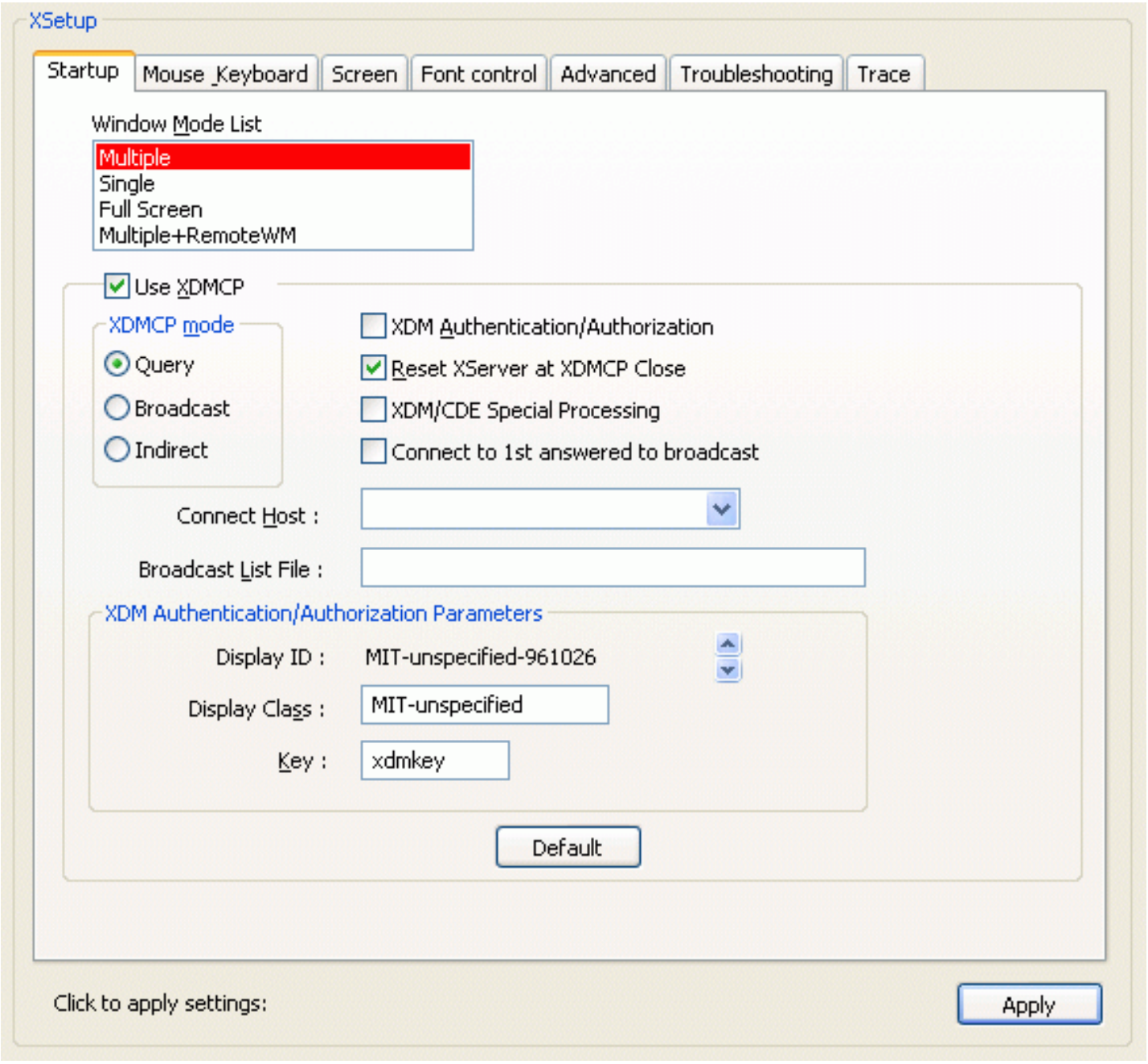
For information about how to edit and restore the registry, view the "Changing Keys and Values" and "Restoring the Registry" Help topics in Registry Editor (in **Regedit.exe**) or the "Add and Delete Information in the Registry", "Edit Registry Data", and "Restoring a Registry Key" Help topics (in **Regedt32.exe**).

Note that you should back up the registry before you edit it. If you are running MS Windows NT/2000, you should also update your Emergency Repair Disk (ERD).

6. Configuring XLitePro **< previous | content | next >**

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

The Startup Tab



The Window Mode List

This item allows you to make a choice of the XServer startup mode. Select a desired window mode by clicking on a mode name. (For more information, see related sections in Chapter **Using XServer**.)

- **Multiple**

In this mode, MS Windows works as a local window manager for your X clients. When an X client starts, it appears in a window like any other displayed by MS Windows. Each client you start creates its own window on your display. The client window's controls (i.e. its borders, the Control Menu box, move window functions, etc.) are all handled by MS Windows on your PC.

- **Single**

  This mode presents all X clients in a single X-session window. Within the window, the window management and all other functions are typically controlled by an X Window System manager you start on a host. The X-session window itself can be sized and moved like any other MS Windows window.

- **Full Screen**

  This mode presents all X clients in a single root window taking up full the screen outside the MS Windows graphical environment. The window management and all other functions are typically controlled by an X Window System window manager you start on a host.

- **Multiple+RemoteWM**

  This mode is the above **Multiple** mode, but the local MS Windows window manager does not control windows of X clients and a user has to run any suitable remote window manager. The mode is very convenient when users use CDE-like interface where a remote window manager provides its own tool/task bar.

The Use XDMCP Box

X Display Manager Control Protocol (XDMCP) is a popular method of starting remote login sessions. Once XServer configured to use XDMCP has initiated X-session for the first time, it contacts an 'xdm' process running on a host system.

XDMCP settings are used to control the XDMCP startup method. The **Use XDMCP** check box lets you specify the XDMCP settings.

Change XDMCP settings only after consulting with your system administrator.

The XDMCP Mode Box

You can check one of the following XDMCP modes:

- **Query**

  In this mode, a particular host used to establish the X connection must be specified in the **Connect Host** field.

- **Broadcast**

  This mode does not allow you to specify a host in the **Connect Host** field. Instead, XServer will broadcast a request to start the X connection to every host named in the **Broadcast List File**.

To use this mode, you must create a **Broadcast List File**.

If you select the **Broadcast** mode, then the **Select XDMCP Host** window will appear after loading XServer. In this window, you will see the XDMCP hosts running on your network, and you can select one of them to start the X-session.

- **Indirect**

The query will be sent to the host specified in the **Connect Host** field. Then this host will either start up or broadcast a request for one or more other hosts to start the X connection.

Connect Host

This field is used to specify a network node name or IP address for the host you want to connect to in the **Query** or **Indirect** startup mode.

When you click on the scroll arrow beside the **Connect Host** box, a drop-down box will display host definitions listed in your **hosts** file. To select a host, just click on an appropriate definition.

If the **hosts** file does not contain the host definition you need, you can enter the host's IP address in the field (in the standard dotted IP address notation).

Broadcast List File

In the **Broadcast** startup mode, specify a file that contains a list of hosts that your PC will transmit a 'broadcast' message to.

The file consists of text lines each of the following format:

IP_address comment
or
name comment

Names must be specified as official host names or aliases in your **hosts** file. Note that the syntax allows you to use the **hosts** file as the **Broadcast List File**.

Note: if you leave this field empty or enter either **0.0.0.0** or **255.255.255.255** (these special destination addresses specify a broadcast), then this will provide the XDMCP broadcast mode (when your PC will transmit a 'broadcast' message to every host on your local network to query all XDMCP daemons).

Reset XServer at XDMCP Close

If checked, this check box enables closing all X clients if the remote XDM daemon terminates the XDMCP session with XServer.

XDM/CDE Special Processing

Check this check box if you are going to use the **CDE XDMCP** mode, i.e. with CDE installed on the remote XDMCP host (for the XServer's **Multiple** window mode only).

Connect to 1st answered to broadcast

This option provides the "old-style" way of using the XDMCP broadcast mode.

XDM Authentication/Authorization

This check box allows you to enable the **XDM-AUTHENTICATION-1** and **XDM-AUTHORIZATION-1** schemes. If your host is using XDM authentication or XDM authorization scheme, set up the values in the **Display ID**, **Display Class**, and **Key** fields and report them to your system administrator.

If you disable XDM authentication/authorization, then XServer will use the default client authorization scheme, **MIT-MAGIC-COOKIE-1**.

Notes:

According to the XSECURITY manual page of X Window System, X provides mechanism for implementing many access control systems. The sample implementation includes some mechanisms with MIT-MAGIC-COOKIE-1 (using shared plain-text "cookies") and XDM-AUTHORIZATION-1 (using secure DES-based private-keys) being two of them.

- **MIT-MAGIC-COOKIE-1**

  When using MIT-MAGIC-COOKIE-1, the client sends a 128-bit "cookie" along with the connection setup information. If the cookie presented by the client matches one that the X server has, the connection is allowed access. The cookie is chosen so that it is hard to guess; xdm generates such cookies automatically when this form of access control is used. The user's copy of the cookie is usually stored in the .Xauthority file in the home directory, although the environment variable XAUTHORITY can be used to specify an alternate location. Xdm automatically passes a cookie to the server for each new login session, and stores the cookie in the user file at login.

  The cookie is transmitted on the network without encryption, so there is nothing to prevent a network snooper from obtaining the data and using it to gain access to the X server. This system is useful in an environment where many users are running applications on the same machine and want to avoid interference from each other, with the caveat that this control is only as good as the access control to the physical network. In environments where network-level snooping is difficult, this system can work reasonably well.

  This system uses 128 bits of data shared between the user and the X server. Any collection of bits can be used. Xdm generates these keys using a cryptographically secure pseudo random number generator, and so the key to the next session cannot be computed from the current session key.
- **XDM-AUTHORIZATION-1**

  Sites in the United States can use a DES-based access control mechanism called XDM-AUTHORIZATION-1. It is similar in usage to MIT-MAGIC-COOKIE-1 in that a key is stored in the .Xauthority file and is shared with the X server. However, this key consists of two parts - a 56-bit DES encryption key and 64 bits of random data used as the authenticator.

  When connecting to the X server, the application generates 192 bits of data by combining the current time in seconds (since 00:00 1/1/1970 GMT) along with 48 bits of "identifier". For TCP/IP connections, the identifier is the address plus port number; for local connections it is the process ID and 32 bits to form a unique id (in case multiple connections to the same server are made from a single process). This 192-bit packet is then encrypted using the DES key and sent to the X server, which is able to verify if the requestor is authorized to connect by

decrypting with the same DES key and validating the authenticator and additional data. This system is useful in many environments where host-based access control is inappropriate and where network security cannot be ensured.

This system uses two pieces of information. First, 64 bits of random data, second a 56-bit DES encryption key (again, random data) stored in 8 bytes, the last byte of which is ignored. Xdm generates these keys using the same random number generator as is used for MIT-MAGIC-COOKIE-1.

Default

This button sets up the default values for check boxes and edit fields in the **Use XDMCP** box.

The XDM Authentication/Authorization Parameters Box

Display ID

If XDM authentication or XDM authorization has been enabled on the XDM host, your system administrator will need to know the value displayed in this field. This field should normally never be changed. The **Display ID** value consists of these two parts separated by hyphen: the **Display Class** value and the arbitrary numerical value.

In very rare cases, your system administrator may determine that your **Display ID** is a duplicate and will ask you to generate a new one. To do this, use arrows on the right side of the **Display ID** field. The up arrow increases the numerical value of the **Display ID**, and the down arrow decreases it.
Do not do this without consulting with your administrator.

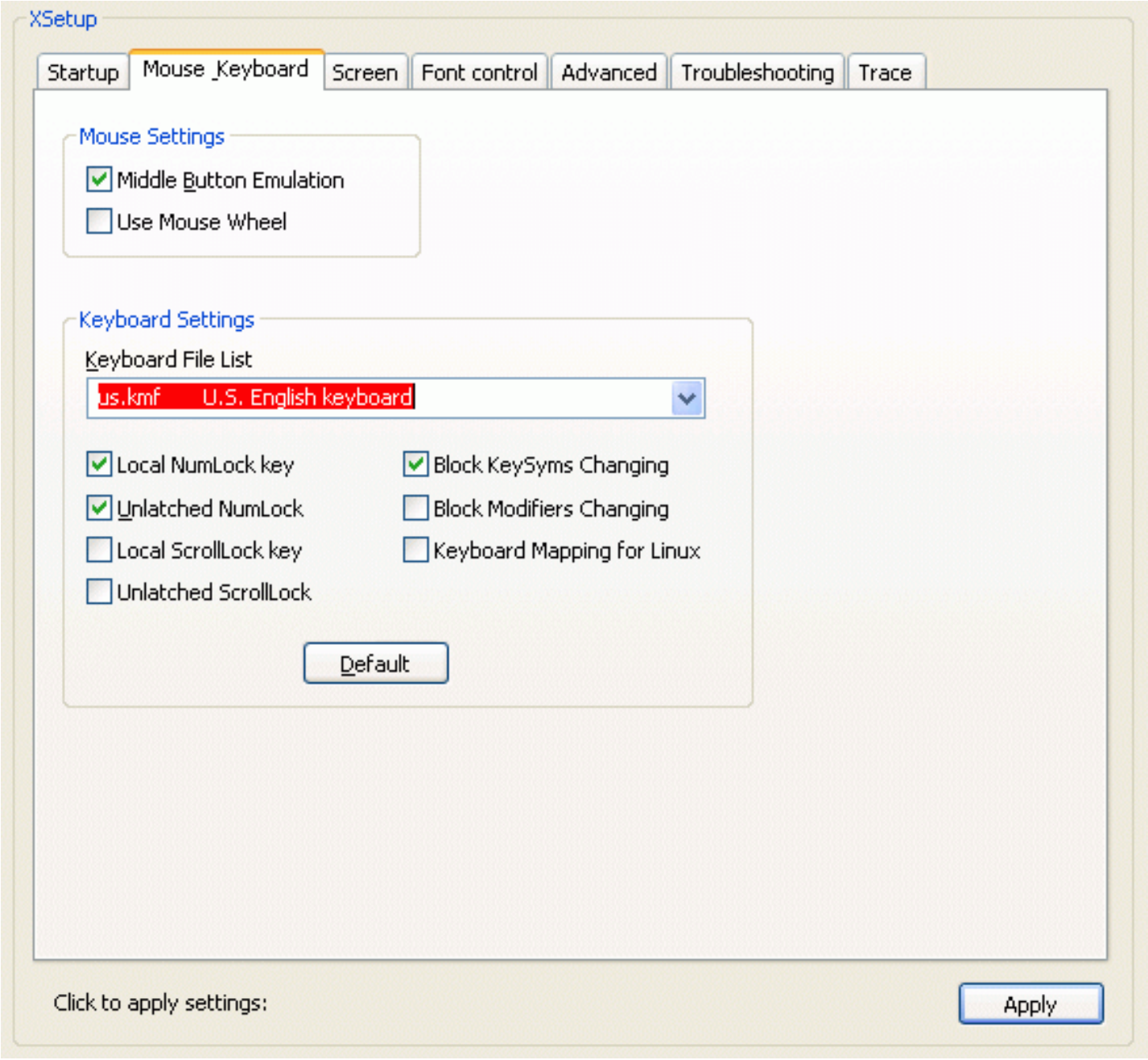
Display Class

This field can be used to group classes of XDM nodes. The field should only be modified at the request of your system administrator. Otherwise it should be left unchanged.

Key

This field defines the key used in XDM authentication. If your host is using XDM authentication, your system manager will need to know the contents of your XDMCP **Key** and **Display ID** fields. This field should only be modified at the request of your system administrator. Otherwise it should be left unchanged.

The Mouse_Keyboard Tab



The Mouse Settings Box

Middle Button Emulation

The default setting specifies a two-button mouse with the check box enabled. If you are using a **3**-button mouse, click to disable the check box. The middle mouse button is emulated by clicking simultaneously both left and right mouse buttons.

Use Mouse Wheel

This check box enables/disables XServer to process the mouse wheel (i.e., to translate its rotation to the Button4 and Button5 press/release X-events).

Note that the mouse wheel does not take effect in the non-maximized **Single Window** mode (because of Scrollbars).

The Keyboard Settings Box

Keyboard File List

You can configure XServer to support different international PC keyboards. The package supplies a set of keyboard mapping files that define assignments of key functions to physical keys on appropriate keyboards. The files are listed in **Appendix A**.

You can enter any kmf-file name in the edit field or select an appropriate keyboard file from the **Keyboard File List** by clicking on its file name. The default keyboard file is **us.kmf**.

For more information, see section **Keyboard Definition Files** in chapter **The XLitePro Database**.

Local NumLock Key

If this check box is enabled, XServer (not X clients) will process the NumLock key.

Unlatched NumLock

If this check box is enabled, XServer will consider the NumLock key as a normal key (non-toggling). The NumLock key is unlatched by default. This was implemented to suppress the NumLock state's influence on some X Window managers and programs.

Local ScrollLock Key

This is important only for the XServer's **Full Screen** mode. The key is used for iconifying the XServer's window. If this check box is enabled, XServer (not X clients) will process the ScrollLock key.

Unlatched ScrollLock

If this check box is enabled, XServer will consider the ScrollLock key as a normal key (non-toggling). The ScrollLock key is unlatched by default.

Block KeySyms Changing

If enabled, this check box prevents the XServer keyboard's KeySyms mapping from external changes (e.g., by the "xmodmap" utility). The default is On.

Block Modifiers Changing

If enabled, this check box prevents the XServer keyboard's Modifiers mapping from external changes (e.g., by the "xmodmap" utility). The default is On.

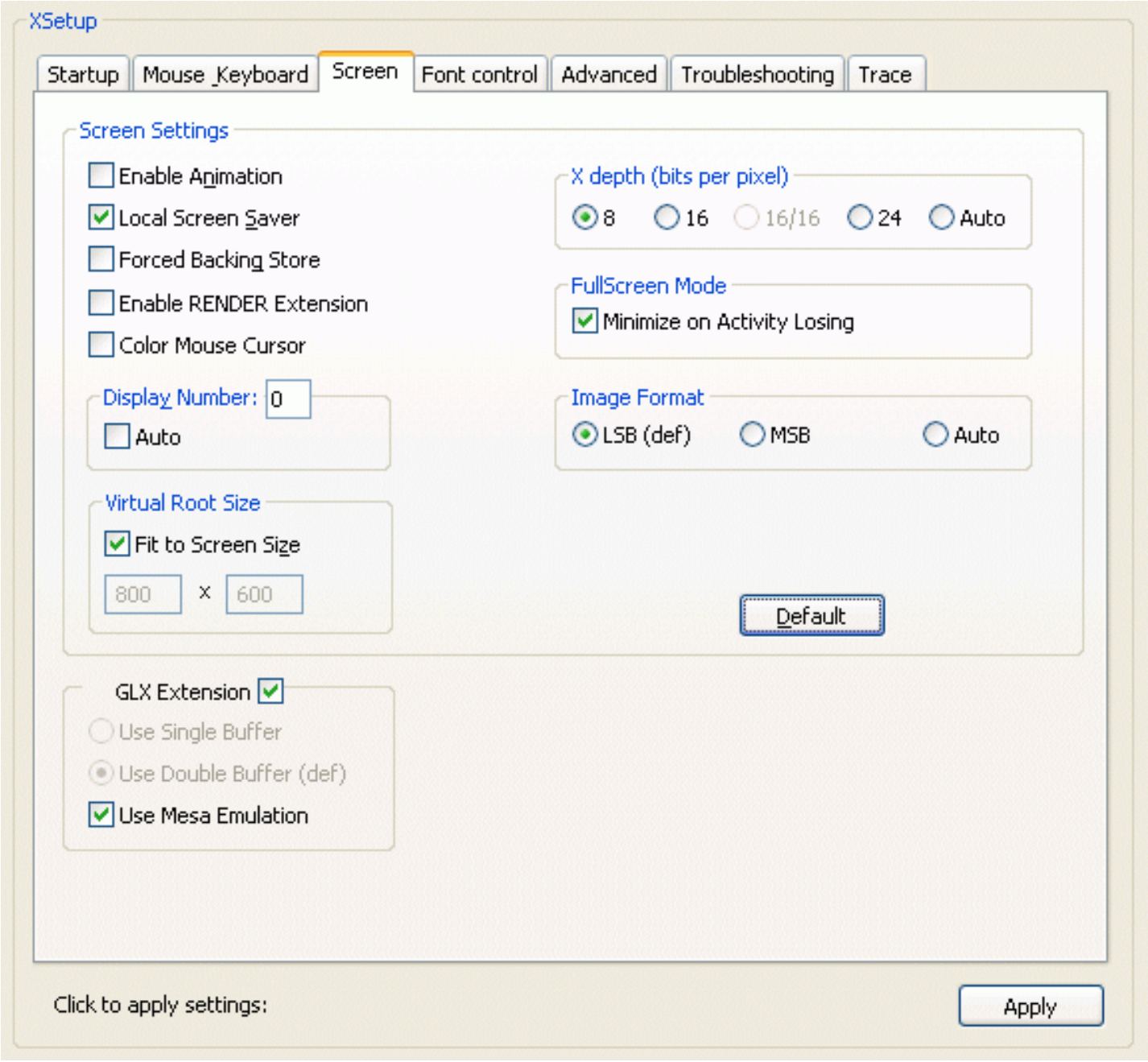
Keyboard Mapping for Linux

If this check box is enabled, XServer will provide a "close-to-Linux" keyboard mapping (i.e., the Linux console keyboard mapping). This setting would be useful when using some applications of the KDE 3.x package (e.g., the "kwrite" editor) that do not correctly recognize some keys (e.g., "Shift +arrows" key combinations).

Default

This button sets up the default values for check boxes and edit fields in the tab.

The Screen Tab



The Screen Settings Box

Enable Animation

If checked, this check box causes XServer to more precisely display color images.

Local Screen Saver

If enabled, this check box causes a Local Screen Saver program to be run (for the XServer's **Full Screen** and **Single** modes only).

Forced Backing Store

This option can be used if the **Backing Store** mode is enabled. (See the **Advanced Tab**.)

If this check box is checked, the **Backing Store** mode will be used with all X clients. The option will cause XServer to use Backing Store on all windows, even if the X-application does not request it.

Certain X-applications will request the Backing Store mode on windows that are complicated to draw. If this check box is clear, XServer will only use Backing Store on those windows that the X-application does request it.

Enable RENDER Extension

This option enables XServer to use the RENDER extension (not available in the 8-bits X depth mode). This also provides a wider range of available pixmap depths.

Color Mouse Cursor

If this check box is checked, then XServer is allowed to use the color mouse cursor when X clients use it.

The Display Number Box

Display Number

You can specify a display number for a particular X-session. This allows you to run simultaneously several X-sessions, each with different **Display Number** (e.g., several Window Managers). Section **Running Several X-sessions** in chapter **Using XServer** describes examples of using the Display Number setting.

According to X11 documentation, from the user's prospective, every X server has a display name of the form: **hostname:displaynumber.screennumber** .This information is used by the application to determine how it should connect to the server and which screen it should use by default (on displays with multiple monitors).

The hostname specifies the name of the machine to which the display is physically connected. For the TCP/IP type of connections, the hostname part of the display name should be the server machine's IP address name. Full Internet names, abbreviated names, and IP addresses are all allowed.

The phrase **DisplayNumber** is usually used to refer to collection of monitors that share a common keyboard and pointer (mouse, tablet, etc.). Most workstations tend to only have one keyboard, and therefore, only one display. Larger, multi-user systems, however, frequently have several displays so that more than one person can be doing graphics work at once. To avoid confusion, each display on a machine is assigned a **display number** (beginning at 0) when the X server for that display is started. The display number must always be given in a display name.

Some displays share a single keyboard and pointer among two or more monitors. Since each monitor

has its own set of windows, each screen is assigned a screen number (beginning at 0) when the X server for that display is started. If the screen number is not given, screen 0 will be used.

Each Display Number corresponds to the known Port Number of XServer (0-6000, 1-6001, etc.).

Note that output log files for different Display Numbers have different names, **xserver[N].out**, with **N** being Display Number.

Auto

This check box enabled activates XServer to dynamically generate Display Number.

In this mode, any new X-session will have new Display Number and a lot of XServer instances (X-sessions) can be started with no changes in the ini-file. This feature is especially useful with NT/2000 Terminal Servers.

Note: no XServer instances must use equal Display Numbers when running simultaneously on your system (even for different users).

The Virtual Root Size Box

When you select either **Single** or **Full Screen** modes for XServer, you can fill in the **Width** and **Height** fields. This lets you set the default size in pixels for the XServer's root window. You can make the virtual screen size larger than your display if you want to (e.g., for multi-monitor systems).

The maximum virtual root size is limited by the expression of "width*height sq. pixels <= 56Mbyte" pixels.

Fit to Screen Size

If selected, this check box allows you to skip input for virtual **Width** and **Height**. XServer will use the values returned by the MS Windows display driver for a single display.

If this check box is clear, you can specify the **Width** and **Height** fields as the default size in pixels that XServer will use for its root window.

If the **Width** and/or **Height** fields are zero or negative values, then the actual screen size will be the sum of the value and the corresponding dimension of the PC's screen. For **Multiple** and **Multiple +RWM** modes, the "0" and "-1" values are only allowed.

The X depth (bits per pixel) Box

These radio buttons let you choose the color depth and visual mode that XServer will use:
- **8-bit** or 256-color visual mode (up to 256 colors);
- **16-bit** or HiColor visual mode (up to 65536 or 32768 colors);
- **24-bit** or TrueColor visual mode (up to 16777216 colors).

The **Auto** choice causes XServer to use current video settings of MS Windows (except for 32-bit, in which case XServer can use up to the 24-bit mode).

Note: if MS Windows is set up to the 8-bit visual mode (256 colors), then XServer will use the same

mode.

The FullScreen Mode Box

- **Minimize on Activity Losing**

  If this check box is enabled, the XServer's window will be iconified each time the focus changes to another window. Otherwise, it can be obscured by other windows.

The Image Format Box

- **LSB**

  This sets up the Image Format to the LSB (Least Significant Byte/Bit first) mode (when every X client must work with XServer using imageByte- and bitmapBit- orders in the LSB format; e. g., X clients on Intel-platform machines).

  This is the default setting (since XServer works with X clients without Image Format conversion).

- **MSB**

  This sets up the Image Format to the MSB (Most Significant Byte/Bit first) mode (when every X client must work with XServer using imageByte- and bitmapBit- orders in the MSB format; e. g., X clients on SUN workstations).

- **Auto**

  This specifies that X clients will define the Image Format for XServer.

Default

This button sets up the default values for check boxes and edit fields in the tab.

The GLX Extension Box

The GLX Extension check box enables XServer to work with X clients that use OpenGL.

XServer can work with a number of X clients simultaneously (in a multi-thread mode of GLX), and X clients may create several GLX windows.

Use Single Buffer

This button allows XServer and X clients to use one buffer for GLX operations (the GLX Single-Buffer mode).

Use Double Buffer

This button allows XServer and X clients to use two buffers for GLX operations (the GLX Double-Buffer mode). This is the default mode.

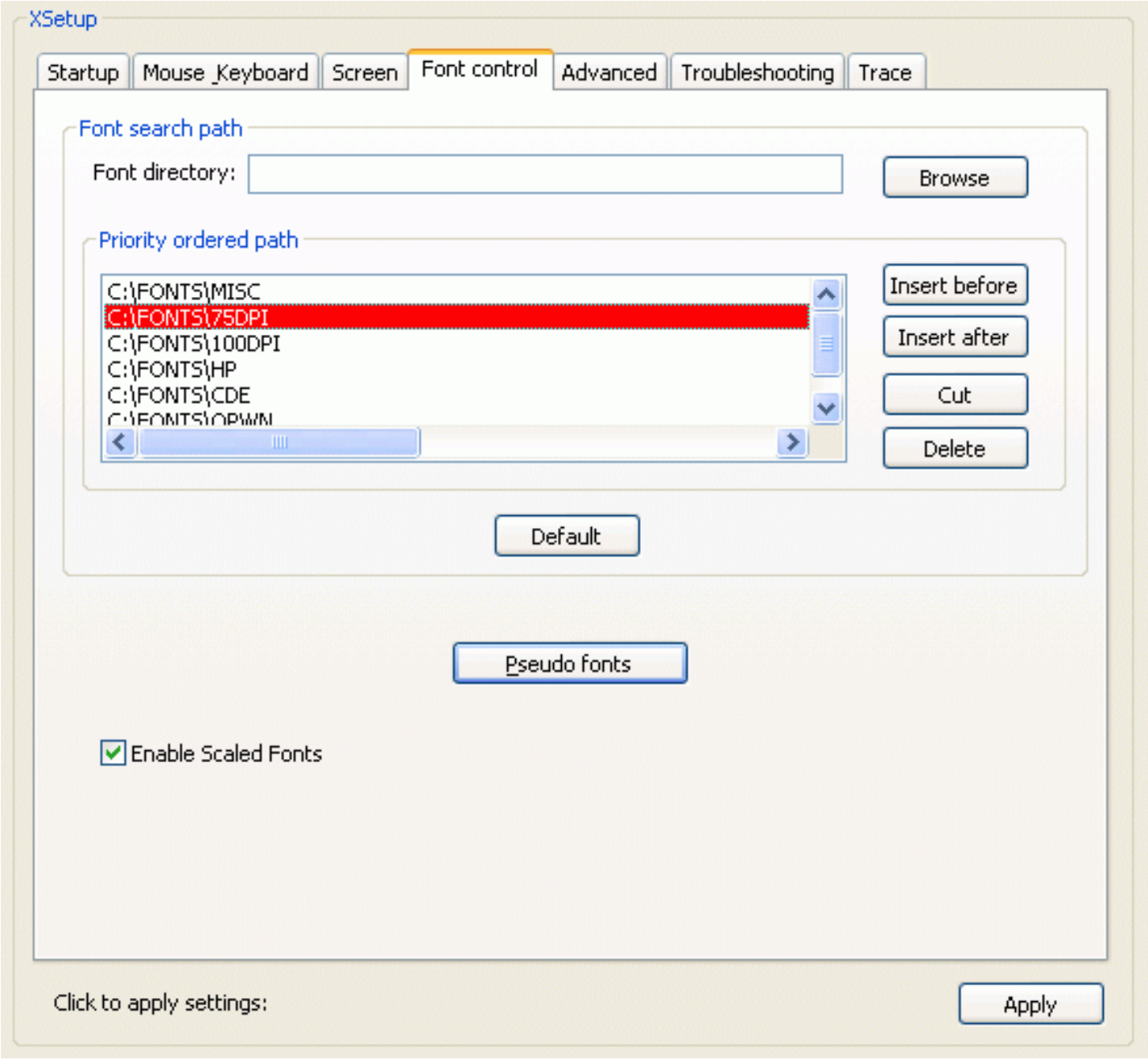
Use Mesa Emulation

This check box allows XServer and X clients to use **Mesa**.

Use the Mesa Emulation mode together with the **Enable RENDER Extension** check box selected.

Mesa is an open-source implementation of the OpenGL specification. OpenGL is a programming library for writing interactive 3D applications. Mesa 5.x supports the OpenGL 1.4 specification. Mesa is used as the core of the open-source XFree86/DRI hardware drivers. Mesa allows OpenGL to be used on systems that have no other OpenGL solution. See www.mesa3d.org for more information.

The Font Control Tab



This tab allows you to manage font sources that can be used in X-sessions and to specify how XServer will use fonts in X-sessions.

Priority ordered path

XServer uses the **Priority ordered path** list on X client's requests for any font. XServer searches for the required font according to the order of font sources till the first matching occurs. In the **Font search path** group box, you can rearrange the **Priority ordered path** list for XServer with the **Insert before**, **Insert after**, **Cut**, and **Delete** buttons.

You can enter a font directory path in the **Font directory** edit field, or use the **Browse** button to select it, or highlight a font path from within the **Priority ordered path** list and press **Cut**.

XServer must have access at least to the **fixed** and **cursor** fonts. The \FONTS\MINIMAL and \FONTS \MISC font subdirectories contain them. So the **Priority ordered path** list must contain these font directories.

You can specify that you want to use a font server running on one or more hosts. Instead of forcing XServer to read all fonts from your PC, the X FontServer Protocol makes it possible to manage fonts separately from XServer, directing XServer to request fonts from a font server.
For example, the **tcp/hp9000:7100** path specifies the font server host called **hp9000** on port **7100**:

For more details, see section **Font Path** in chapter **Font Control**.

Enable Scaled Fonts

If this check box is checked, then it allows XServer to use scaled fonts.

Default

This button sets up the default values for check boxes and edit fields in the tab.

Pseudo fonts

If you plan to use MS Windows fonts in X-sessions, press the **Pseudo fonts** button to choose MS Windows fonts and assign aliases to them. This means creating pseudo fonts.

MS Windows font specifications and aliases for Pseudo Fonts are stored in the Pseudo Fonts Directory. The Pseudo Fonts Directory, WINFONTS subdirectory, is located under the home directory (by default) of the software package.

If you need some X font that is inaccessible, XServer allows you to use any MS Windows font instead of it. MS Windows fonts do not support X11 font naming conventions. Therefore, pseudo names (aliases) are used to access them. To use a MS Windows font, you should create a pseudo font for it (i.e. choose one and assign a desirable alias). Pseudo fonts can be created in advance or immediately in the X-session.

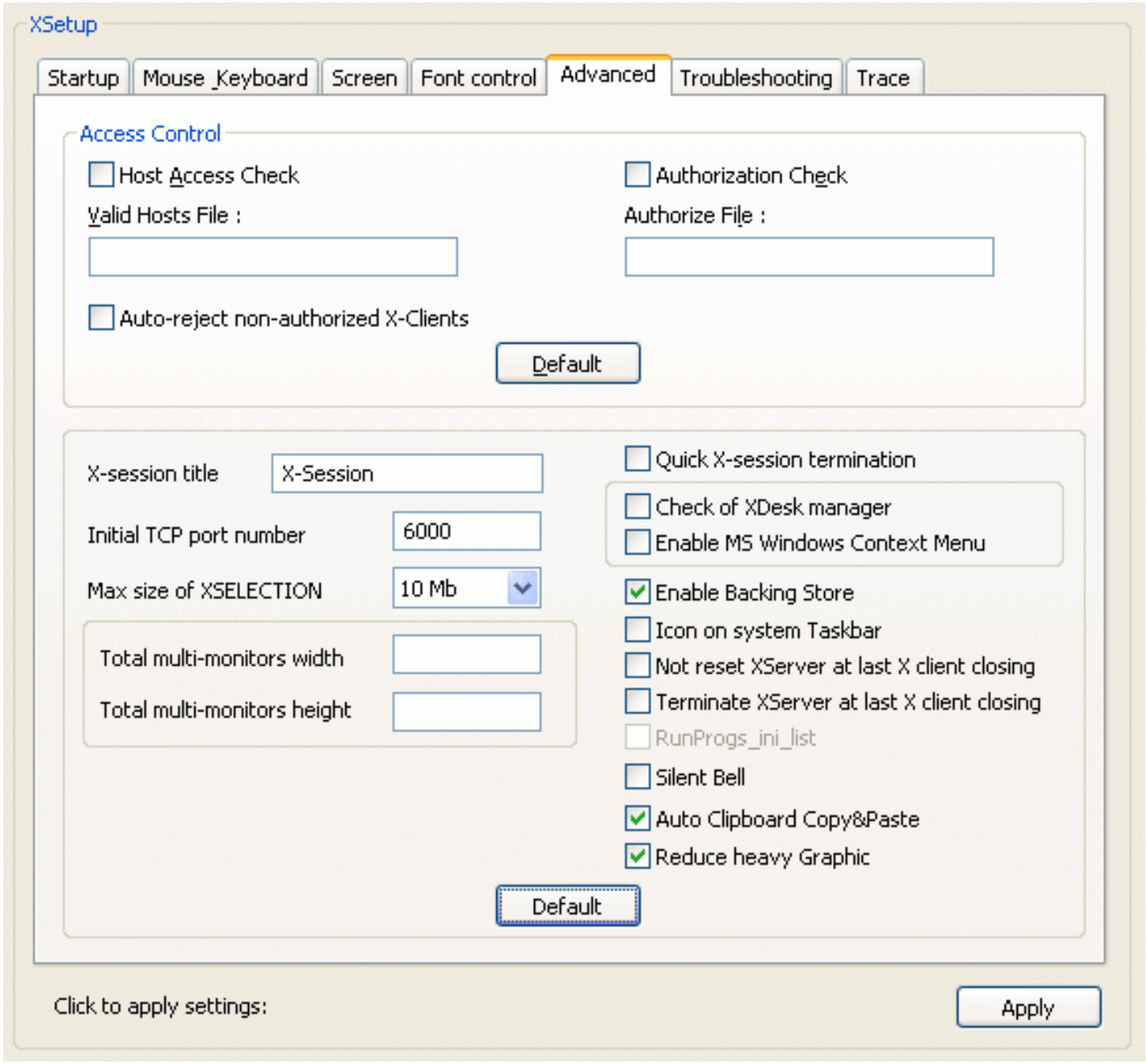
Loading pseudo fonts takes some time to create their images. Reading font image from the X11 format file is faster. It becomes essential if X client uses a lot of fonts. To prevent this time loss, you can compile and save images of frequently used pseudo fonts in the X11-like format in your Pseudo Fonts Directory or in any font directory for later use. This will require some disk space.

You can also create a pseudo font directly in the X-session on particular X client's font requests. Unfortunately, pseudo fonts created on X client's requests become accessible only after restarting the X-session. In the current X-session, XServer uses the default font instead of requested one.

You can define whether to use pseudo fonts in the X-session or not. You should define priority of Pseudo Fonts Directory among other font sources in the **Priority ordered path** list. XServer will use pseudo fonts as well as X fonts.

On how you can manage pseudo fonts, see section **Pseudo Fonts** in chapter **Font Control**.

The Advanced Tab



X-session title

In this entry field, you can specify your title for an X-session (and its icon) instead of its default title. **Note**: A full title is the X-session title followed by its virtual DisplayNumber.

Initial TCP port number

In this entry field, you can change the default (6000) initial port number for your X-session that

XServer will use.

Max size of XSELECTION

This option is used to change the default (1MB) size for X-Selection that may be copied to/from the MS Windows' Clipboard (in Bytes).

Total multi-monitors width
Total multi-monitors height

With these two values, you can set the forced multi-monitors mode if XServer cannot detect the multi-monitors mode automatically.
WWW is the sum width and the HHH is the sum height of your "monitors".
In the **Full Screen** mode, XServer can only use the MAIN monitor.

Quick X-session termination

This option is used to prevent from displaying the confirmation message on closing X-sessions.

Check of XDesk manager

This option is used to provide the correct focus processing by the XDesk virtual desktop manager.

Enable MS Windows Context Menu

This option is used to enable the standard MS Windows Context Menu (when right-clicks on the window's title bar provide the standard drop-down menu).

Enable Backing Store

This option lets you enable XServer to process the **Backing Store** mode. See also **Forced Backing Store** on the **Screen Tab**.

If this check box is checked, the **Backing Store** mode will be used with all X clients. This allows X Window System displays to be saved off-screen so that X clients do not have to refresh a window display after it has been obscured by another window. Saving a copy of the information obscured when windows overlap each other reduces network traffic. Certain X-applications will request the Backing Store mode on windows that are complicated to draw. The option will cause XServer to use Backing Store on all windows that the X-application requests it.

If this check box is clear, the Backing Store mode will be turned off so XServer will not save copies of windows information, even if the X-application does request it.

Icon on system TaskBar

This option enables your system to keep the X-session icon (as well as icons of applications started from the XServer's Run menu) in the system taskbar (for the **Multiple** modes).

Not reset XServer at last X client closing

This option is used to prevent XServer from resetting after the last X client closes its session (for not

removing those resources that have been created by X clients run).

Terminate XServer at last X client closing

This option is used to automatically terminate XServer after the last X client closes its session.

Silent Bell

When enabled, this check box will block all sounds from all X applications of the package (XServer, X Clients, etc ...) (i.e., internal sounds and the X-protocol XBell requests).

Auto Clipboard Copy&Paste

This check box controls the mode of connection between the MS Windows Clipboard and the current X Selection. When enabled, then changing of the MS Windows Clipboard contents will force the same change in the current X Selection. Also, changing of the contents of the current X Selection (from the active application window) will automatically cause copying it into the MS Windows Clipboard (so you are ready to paste).

Reduce heavy Graphic

If this check box is selected, then XServer will use some methods to speed-up drawing (e.g., lossy drawing of images).

Default

This button sets up the default values for check boxes and edit fields in the box.

The Access Control Box

This box contains the following items that allow you to restrict access to your XServer from remote hosts. You can give access only to the hosts you authorized in the **Authorize File** and/or you specified in the **Valid Hosts File**.

Host Access Check

If this check box is checked, then XServer will check host access by using a file you specify in the **Valid Hosts File** field.

If this check box is disabled, then XServer will not check host access, so every host on your network will have access to your XServer (and the **Authorization Check** state does not matter).

Valid Hosts File

Use this field to specify a file you created to give access to hosts (and X clients) you wish to connect to your XServer.

This file will be used only if the **Host Access Check** check box is checked.

The file consists of text lines each of the following format:

IP_address comment
or
name comment

IP addresses are specified in the dotted IP address notation. Names must be specified as official host names or aliases in your **hosts** file.

Note that the host definition syntax allows you to use your **hosts** file as the **Valid Hosts File**.

Authorization Check

If this check box is checked, then XServer will make standard authorization check by using a file you specify in the **Authorize File** field.

If this check box is disabled, then XServer will not make standard authorization check.

Authorize File

Use this field to specify a file you created to give access to hosts you wish to connect to your XServer (for either XDM and non-XDM clients).

This file will be used only if the **Authorization Check** check box is checked.

Note: to use the **Authorize File**, you should copy the standard authorization binary file, **~HOME/.Xauthority**, created with the 'xauth' utility on your host.

Auto-reject non-authorized X clients

If this check box is checked, then XServer will suppress the Authorization Audit message. This setting is useful if you want to reject any unauthorized X clients automatically (with no confirmation dialog).

Default

This button sets up the default values for check boxes and edit fields in the **Access Control** box.

Some Hints about Authorization and Host Access Policy

XServer checks permissions of hosts (and X clients) to establish connection by the following rules:

1. XServer will (firstly) make standard authorization check (if the **Authorization Check** check box is checked) and will (secondly) check host access (if the **Host Access Check** check box is checked).

   So if the **Host Access Check** check box is disabled, then every host on your network will have access to your XServer.

2. If the **Authorization Check** check box is enabled, and a host is found in the **Authorize File**, then XServer will give access to the host.

3. If the **Host Access Check** check box is enabled, and a host is found in the **ValidAccessFile**,

then XServer will give access to the host.

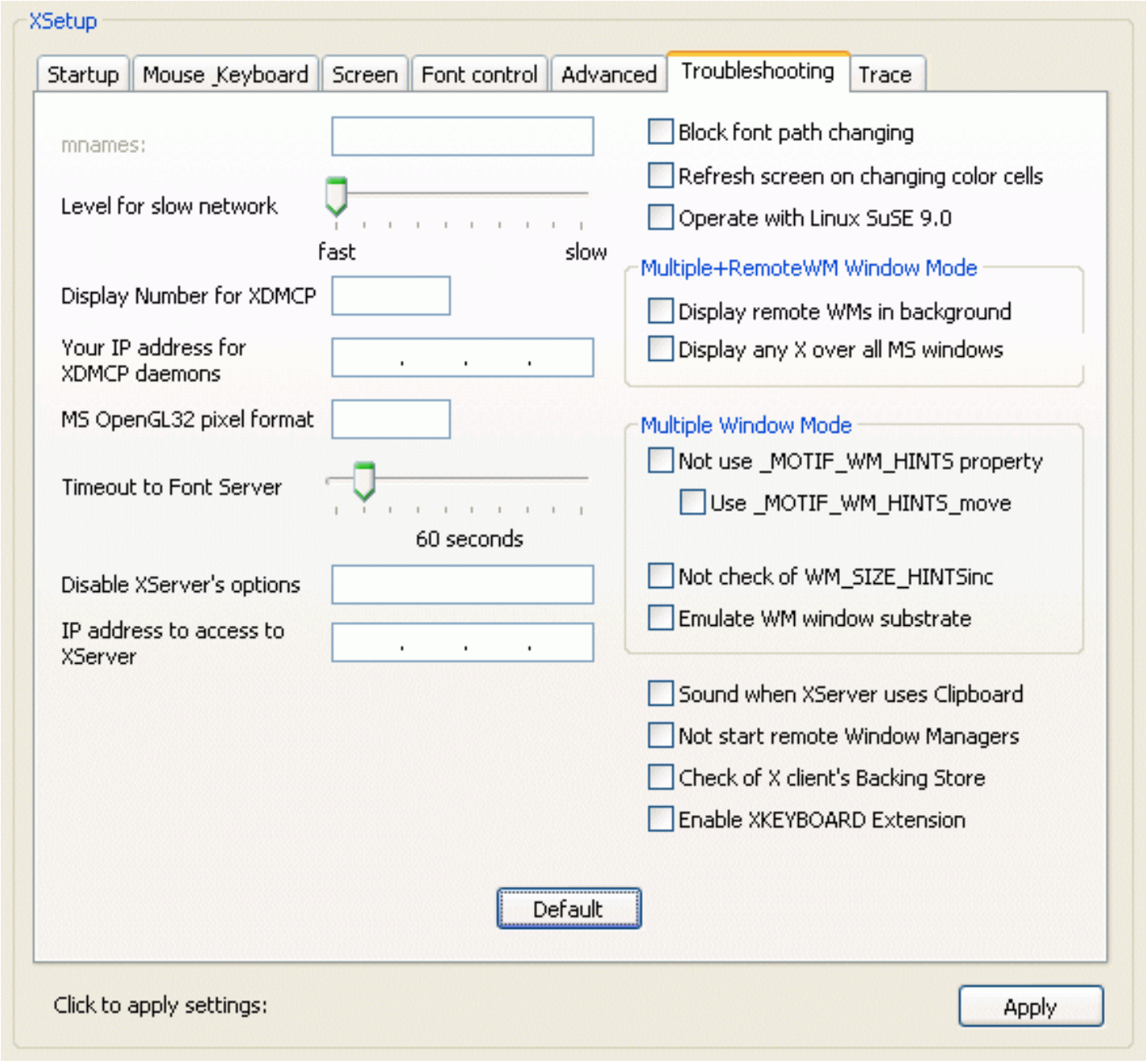
4. If the **Host Access Check** check box is enabled, but the **ValidAccessFile** field is empty or the file you specified in the field cannot be normally read or the file is empty, and the **Authorization Check** check box is disabled, then access to your XServer is disabled for all hosts.

5. To authorize hosts that are absent in the **ValidHostsFile**, you can enable the **Authorization Check** check box and specify the **Authorize File**.

6. Configuring XLitePro **< previous | content | next >**

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

The Troubleshooting Tab



Options that you can specify within this tab are mainly for debug purposes.

Level for slow network

This option enables XServer to avoid the focus-problem when your network (or X client's machine) is slow, and XServer processes the "ButtonRelease" mouse event before it receives the "AllowEvents"/"GrabKeyboard" requests from the corresponding X client.

A slider position defines a priority level of kbd/mse events when redrawing an image that is associated with the mouse pointer position or key. For slow network or X client's machine, it may be preferable to reduce traffic due to reducing mouse pointer positions transmitted to X client and so reducing image redrawing.

Display Number for XDMCP

This option enables you to provide a "virtual" value of DisplayNumber for XServer. This value is sent to a remote XDMCP daemon instead of the real value.

Note: These settings are necessary for some programs running on remote computers to communicate with your PC from outside proxy (e.g., for VPN and other proxy-like programs).

Your IP address for XDMCP daemons

This option enables you to specify an IP address that XServer will use to communicate with the XDMCP daemon.
This is applicable when the XServer's PC has more than one IP address. In this case, XServer will use the IP address specified.

MS OpenGL32 pixel format

An integer value in this entry field specifies an appropriate pixel format used by MS Windows OpenGL32 (not in the **MESA** mode; see **GLX Extension** on the **Screen Tab**).

If you enter 0 or leave the field empty, then MS Windows OpenGL32 will use MSDN function ChoosePixelFormat. The ChoosePixelFormat function attempts to match an appropriate pixel format supported by a device context to a given pixel format specification.

If you enter a positive integer, then MS Windows OpenGL32 will use the value to match an appropriate pixel format to a given pixel format specification.

If you enter a negative integer, then MS Windows OpenGL32 will use MSDN function DescribePixelFormat to find an appropriate pixel format. The DescribePixelFormat function obtains information about the pixel format.

Timeout to Font Server

This slider changes the default timeout value used for access to remote font servers.

Disable XServer's options

This entry field disables items from the XServer's Control menu:
Restore, Move, Size, Minimize, Maximize, Close,
Edit, Options, Refresh, LocalRef, Restart, Macros, Run, Messages,
About, Help, LocalWin, Readme.

To disable some of the menu items, enter them in the field comma separated.

IP address to access to XServer

This option enables you to bind an IP address to the XServer's listening socket.

E.g., the **127.0.0.1** value allows access to XServer from local X clients only.

Block font path changing

If disabled, this check box prevents from changing the XServer Font Path by external requests (so remote X clients are not allowed to change Font Resources). The default is OFF.

Refresh screen on changing color cells

This option is used to set the immediate refresh of screen after any color cell of the active colormap is changed.
Be ready that this setting significantly slows down the XServer's performance.

Operate with Linux SuSE 9.0

This option enables XServer to be operational with Linux SuSE 9.0.

This option is relevant to the **TrueColor** visual mode with the **Enable RENDER Extension** check box cleared. In this case, XServer will "emulate the 8-bit pixmaps creation" allowing X applications (running under Linux SuSE 9.0) to use 8-bit pixmaps. To enable RENDER Extension is the most correct way.

The Multiple+RemoteWM Window Mode Box

Display remote WMs in background

This option is used to allow a remote window manager to display in background if it provides this (e. g., olwm does not provide, but dtwm does). Such a background window is suppressed by default because it covers all non-X windows.
This setting is for the XServer's **Multiple+RemoteWM** mode.

Display any X over all MS windows

This option allows XServer to display any X windows on top of MS windows. This setting is for the XServer's **Multiple+RemoteWM** mode. In this case, it is convenient to use local mwm.

The Multiple Window Mode Box

Not use _MOTIF_WM_HINTS property

This option enables XServer to prevent from processing the "_MOTIF_WM_HINTS" windows property (that may be requested by remote X clients).

- **Use _MOTIF_WM_HINTS_move**

  This option enables XServer to prevent from processing the "_MOTIF_WM_HINTS_move" windows property (that may be requested by remote X clients).

Not check of WM_SIZE_HINTSinc

This option enables XServer to prevent from processing the "WM_SIZE_HINTS" windows property (that may be requested by remote X clients).

Emulate WM window substrate

If this check box is selected, XServer will emulate a window substrate of a remote Window Manager (for some Client applications that require it to work properly).
This is a special mode for some non-standard X11-Clients which require the existance of a parent window for the Client's window as a sign of running Window Manager.

Sound when XServer uses Clipboard

This check box enables system bell sound when XServer will perform Edit operations (i.e., from the Edit menu) with Clipboard.

Not start remote Window Managers

If this check box is selected, XServer will disable starting a remote window manager in answer to its request since there is another window manager already running (for **Single** and **Full Screen** modes). In the **Multiple** modes, only local window managers are allowed to run.

Check of X client's Backing Store

This option is only used in the **Multiple** and **Multiple+RemoteWM** modes.

If this check box is enabled, XServer will use the Backing Store mode, when the X-application requests it. But XServer will not use Backing Store while re-displaying an X-application's window in some cases.

If this check box is clear, XServer will use the Backing Store mode, when the X-application requests it.

To try this option may be useful when an X client uses the Backing Store mode and the windows are not properly redrawn.

See also **Enable Backing Store** on the **Advanced Tab**.

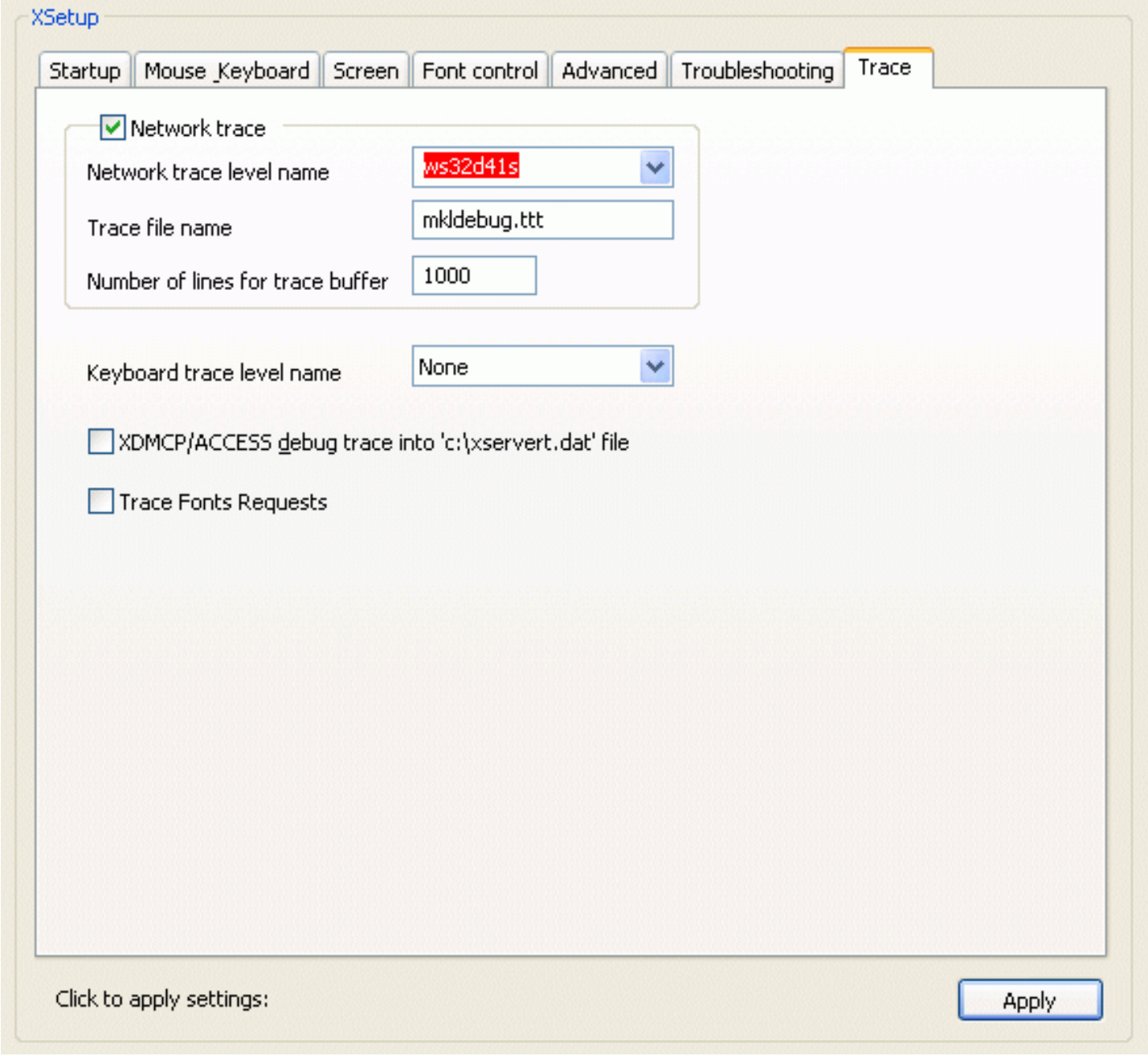
Enable XKEYBOARD Extension

If this check box is selected, XServer will use standard XKEYBOARD Extension (for some Client applications that require it to start up properly).

Default

This button sets up the default values for check boxes and edit fields in the tab.

The Trace Tab



Options that you can specify within this tab are for debug purposes. They exert influence on all X applications of the package (XServer, X Clients, etc ...).

Keyboard trace level

A value (one from **None**, **Minimal**, **Advanced**, **Full**) in this field denotes a tracing level for collecting keyboard trace information for your X-session (e.g., Events of pressing/releasing of keys) to the log file.

None disables keyboard tracing.

XDMCP/ACCESS debug trace into 'c:\xservert.dat' file

If checked, this check box enables writing out any special debugging information into the **c:\xservert.dat** trace file.
Be careful when using this option because of significant decreasing the XServer's performance and a large size of the trace file.

Trace Fonts Requests

If this check box is checked, then all font requests from X clients (with resolve messages) will be stored in the **xserver.out** file. This option is useful for analysis of the font accessibility and resolving font problems by XServer.

The Network trace Box

When the **Network trace** check box is selected, it enables network tracing between XServer and remote X clients.

Network trace level name

A value in this field that you can choose from the list box denotes a tracing level for collecting network trace information for your X-session to the log file.

Trace file name

In this entry field, you can specify a name for a log file to store network trace data.

Number of lines for trace buffer

In this entry field, you can type in a length for the allocated memory (in number of lines) to flush it to the trace file.
The value of 0 means a "very large" buffer (as much as your system allows).
Caution: when a system crash happens, the data in the allocated memory is lost.

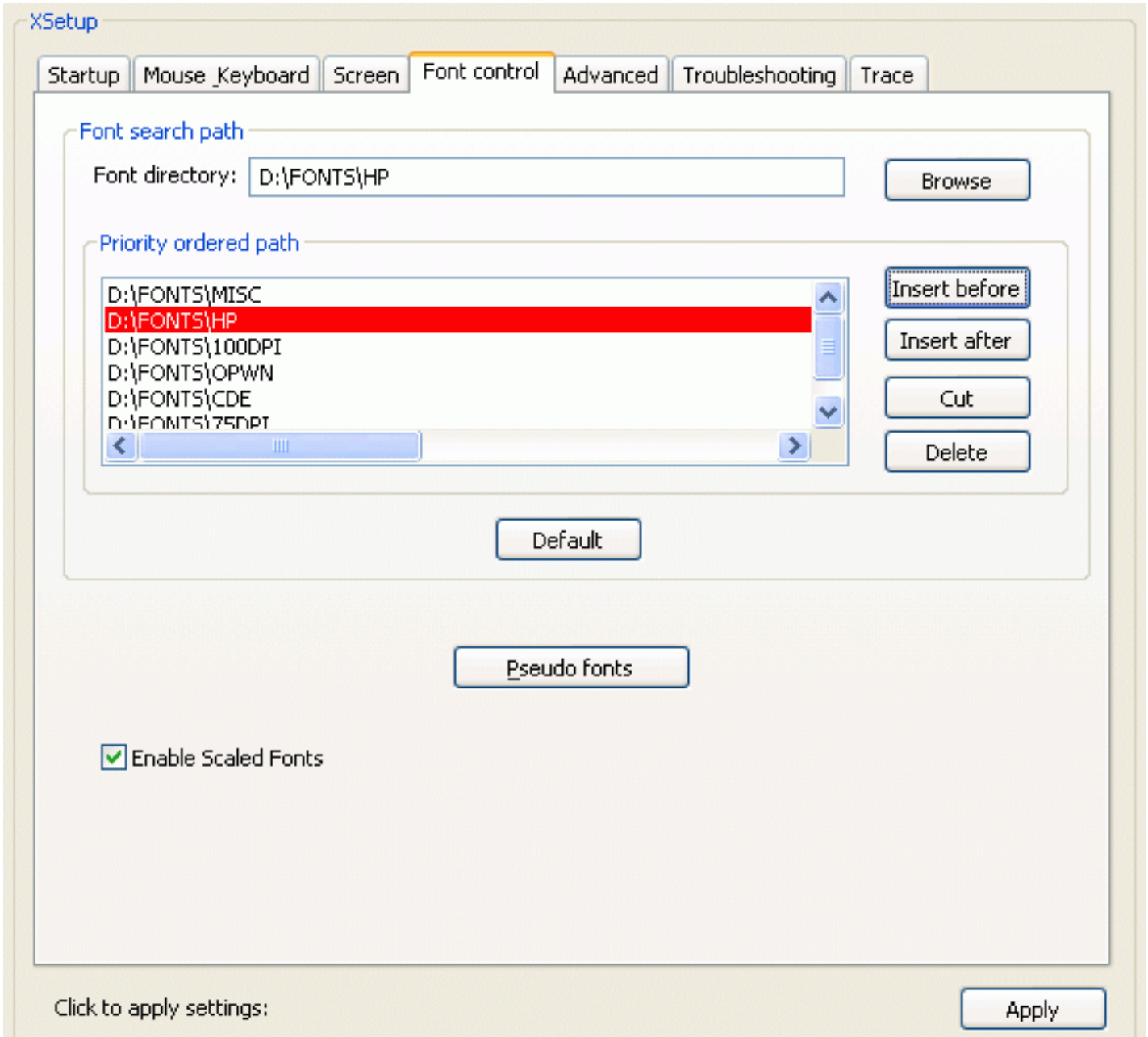
Font Path

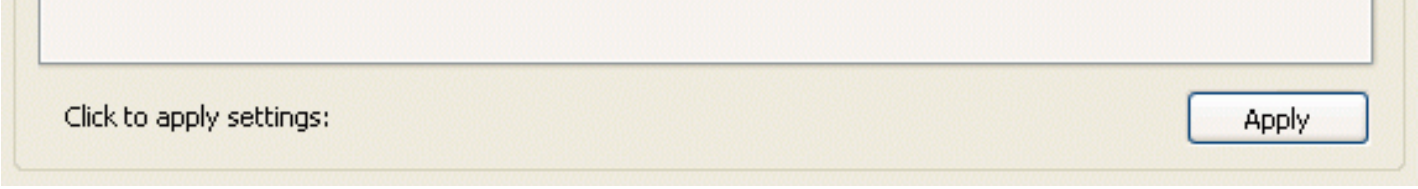
Font Path is an ordered table of font sources (a **Priority ordered path** list). A font source is one of the following: a font directory, a reference to a remote font server, and a pseudo font directory. XServer uses Font Path on X client's requests for any font. XServer searches for the required font according to the order of font sources till the first matching occurs.

XLitePro provides wide range of local X fonts (standard font sets supplied by MIT along with X11 R6). These font sets are immediately available after installing the package. They appear in Font Path as font directories under the FONTS subdirectory in the XLitePro home directory.

Also, XServer can use different locale 16-bits fonts including Chinese, Japanese, or Korean fonts, etc.

To manage Font Path, choose **XSetup** and then select the **Font Control** tab in the **Settings** window.





Note: The FONTS subdirectory shown in pictures of this chapter is not under the XLitePro home directory. Users can move it to a desired location in their file systems (see section **Editing Font Path** below) after installation of the package.

The **Apply** button in the tab writes new Font Path into the **xwp.ini** file located in the XLitePro configuration directory.

If the **Enable Scaled Fonts** check box is selected, it allows XServer to use scaled fonts.

Font Directory

A font directory contains a number of font description files in the X11 format. Setup installs a set of font directories under the FONTS subdirectory in the XLitePro home directory (by default).

Any font directory must have the **fonts.dir** file (that contains the font description list according to font files of the directory), the **fonts.ali** file (with font aliases), and a set of font description files (with the **.pcf**, **.snf**, or **.pgz** file name extension).

The **fonts.alias** file name as well as **fonts.ali** can be used in the FONTS directories.

Note: XServer can only use uncompressed **.pcf** (X11R5 or later), **.snf** (X11R4), and **.bdf** files. Also, XServer can use these font files compressed with the **compress** or **gzip** utility (for X11R6.3 only).

If your remote X server supports 'compressed' fonts (X11R6 server does. X11R6.3 server supports 'gzipped' fonts too), you may compress the **.pcf** and **.snf** font files with the '**compress**' or '**gzip**' utility (for X11R6.3 only).

You have to convert each **.bdf** file to **.pcf** (X11R5 or later) or **.snf** (X11R4) with the **bdftopcf** and **bdftosnf** utilities.

To convert each **font.bdf** file in the set, run the following command line:

- in X11R5, **bdftopcf font.bdf > font.pcf**
- in X11R4, **bdftosnf font.bdf > font.snf**

XServer must have access at least to the **fixed** and **cursor** fonts. The \FONTS\MINIMAL and \FONTS \MISC font subdirectories contain them. So Font Path must contain these font directories at least.

Font Server

You can specify that you want to use a font server running on one or more hosts. Font servers are defined in the X11 R6 release of the X Window System. Instead of forcing XServer to read all fonts from your PC, the X FontServer Protocol makes it possible to manage fonts separately from XServer, directing XServer to request fonts from a font server via the X Consortium standard network protocol. In addition, for fonts that take a long time to open, this allows XServer to continue with other clients while the font server services the font requests.

A font server specification for TCP/IP has the following format:

tcp/name: port [/catalogue+catalogue+...]

name

The network name or IP address for the host that the font server is running on.

port

The remote port that the font server is listening on. The port number value is usually of 7100 or 7500 depending on the operating system used, but check this with your system administrator.

catalogue

An optional list of the font catalogue(s) you want to use. If more than one catalogue is specified, separate each name with a 'plus' sign (+).

For example, the following specifies the font server host called **hp9000** on port **7100**:

tcp/hp9000:7100

You can include more than one font server specification into the Font Path list.

Note: You must add an entry (with the host name and IP address for the remote font server you want to use) into the **hosts** file used by MS Windows:

- **C:\WinNT\System32\drivers\etc\hosts** for MS Windows NT4/2000/XP
- **C:\Windows\hosts** for MS Windows 9x/ME.

Editing Font Path

To remove a highlighted font source from Priority Ordered Path, press **Delete**.

To move a highlighted font source from Priority Ordered Path to the **Font directory** edit field, press **Cut**.

To insert the **Font directory** edit field into Priority Ordered Path, select a path from within the **Priority ordered path** list and then click **Insert before** or **Insert after**.

Also, to fill in the **Font directory** edit field, you can manually enter a font directory path or use the **Browse** button to select it (i.e. a path to a **fonts.dir** file).

Pressing **Default** places font paths in the **Priority ordered path** list according to a default order.

You can use the 'minus' and 'space' characters in the Font Path items. The total length of the Font Path items can be up to 2040 characters.

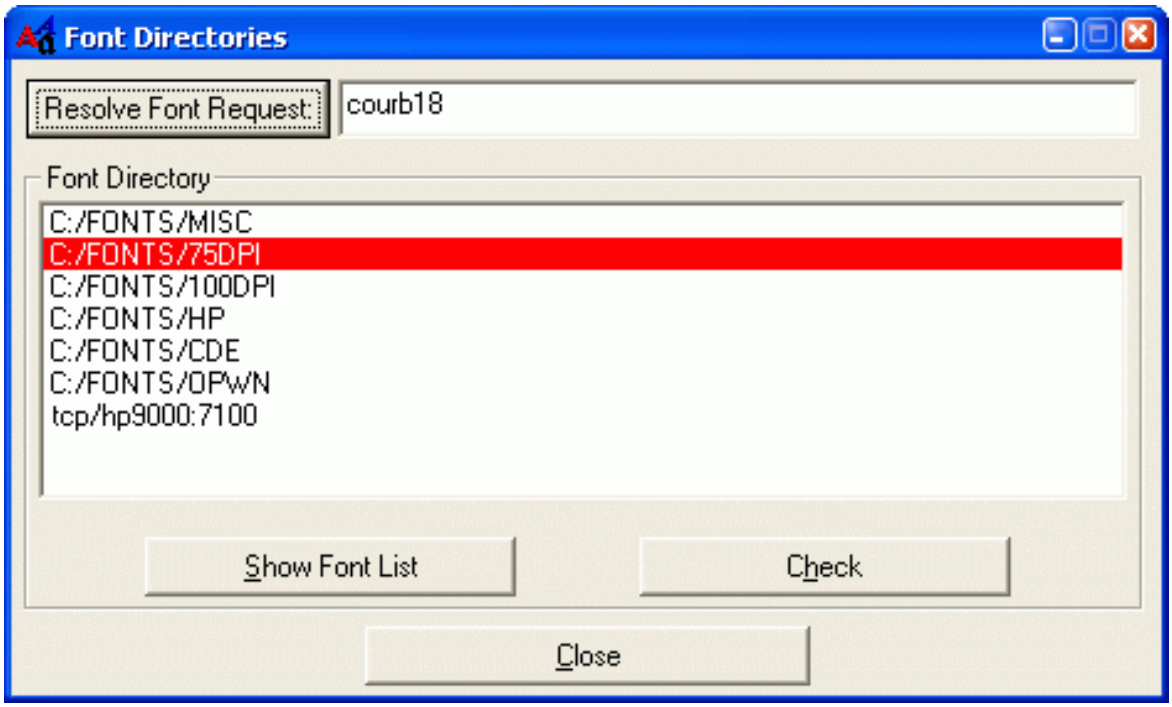
Also, see **Appendix C Troubleshooting** for examples.

Font Service

The **XFontset Service** utility allows you to view fonts accessible for X clients in the current X-session. You can use this option instead of running the **xlsfonts** and **xfd** X clients.

When you click on the **XFontset Service** icon in the XLitePro Programs' folder,

the **Font Directories** dialog box appears:



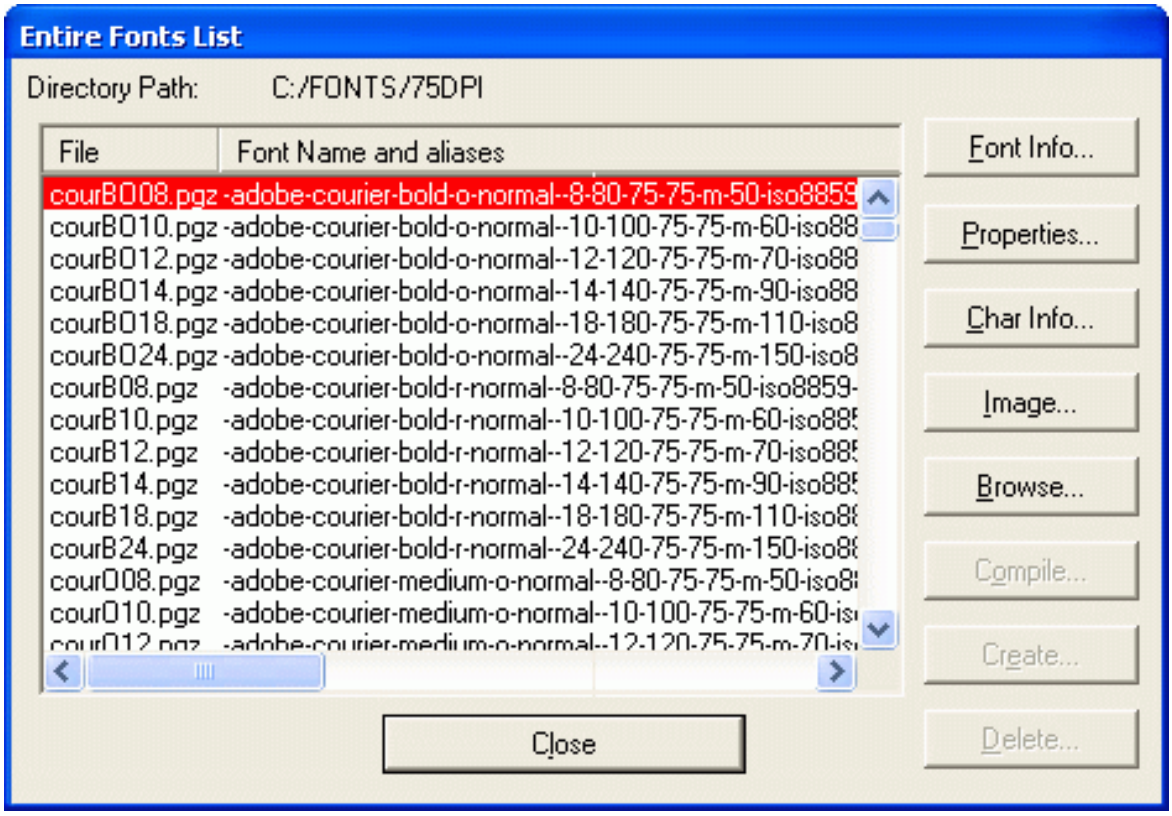
Also, to display this box, you can run **Font Manager** from the X-session Control menu.

This dialog box shows font directories existing in Font Path (i.e. a **Priority ordered path** list).

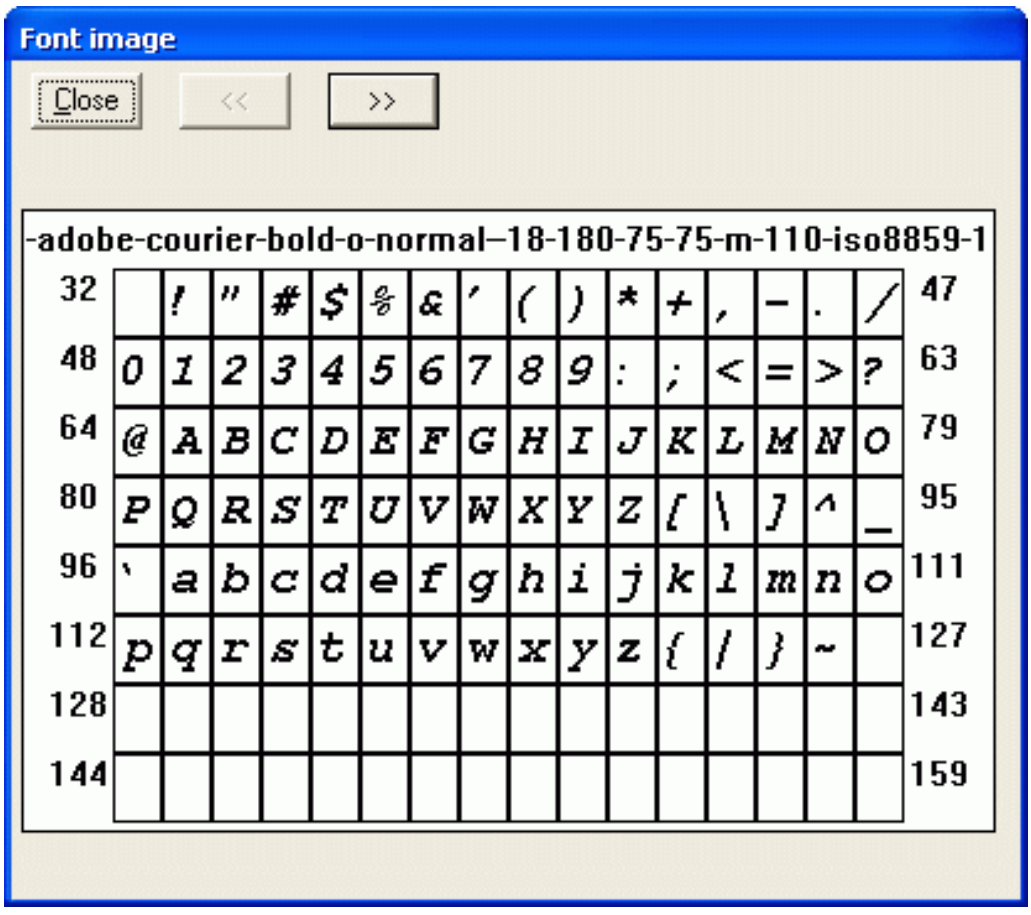
You can check how XServer resolves a particular X client's font request by typing the requested font name in the edit field and then pressing the **Resolve Font Request** button. The program will show you the font directory and the font file corresponding to the requested name. You can use the 'asterisk' sign (*) as the wildcard character in the edit field.

You can check data consistency of any highlighted font directory by pressing **Check**. The program will tell you about detected unresolved font aliases and unresolved references to font files in this font directory. This might be useful after you made any manual changes in the font directory.

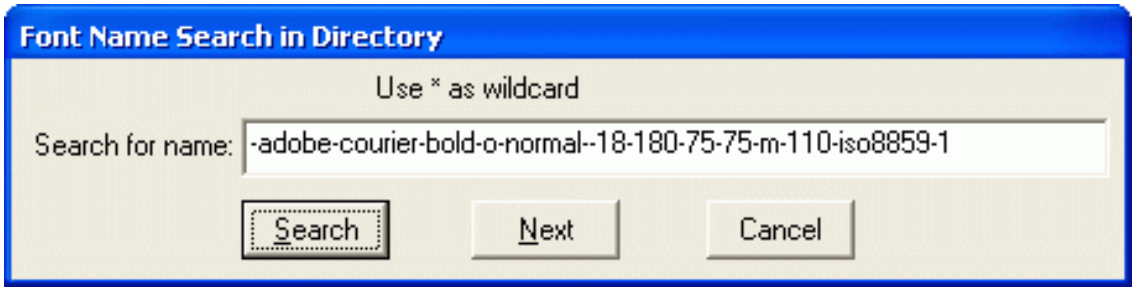
You can get all font names existing in any font directory by highlighting it and then pressing **Show Font List**, or double-clicking on a font directory. The **Entire Fonts List** dialog box will appear.



This dialog box shows font files and respective font names and aliases (from the **fonts.dir** file). For any font highlighted, you can view (in Notepad windows) **Font Info**, **Properties**, **Char Info** (and save these text files). Also, pressing **Image** will display a box like below with character images of the selected font.



To find a particular font name in a long font list (and then to view font info), you can use the **Browse** button, and then press **Search** in the dialog box that will appear.



Note: You cannot view a font that is shaded by a similar font name in the font directory preceding it in Font Path. To view and use such a font, you should place its directory first in Font Path.

Understanding the Font List

You can list the fonts available on your remote system by typing the following command in an X window:

```
xlsfonts | more
```

A list appears displaying one font per line. This "font name" is typical:

```
-adobe-courier-bold-o-normal--10-100-75-75-m-60-iso8859-1
```

The hyphen-separated fields designate the following:

- `adobe` **Foundry**
- `courier` **Family**
- `bold` **Weight**
- `o` **Slant**
- `normal` **Set Width**
- **Additional Style**
- `10` **Pixel Size**
- `100` **Point Size**
- `75` **Y Resolution**
- `75` **X Resolution**
- `m` **Spacing**
- `60` **Average Width**
- `iso8859` **Character Set Registry**
- `1` **Character Set Encoding**

The most important properties to consider are the **Family**, **Weight**, **Slant**, and **Point Size**.

Family is the style of the letters. Some common examples are **Helvetica**, **Courier**, and **Times**.

Weight is the width of the line making up the letter. The options are **regular** and **bold**.

Slant is the angle of the letters. The most common are **R** for Roman (meaning upright), **I** for italic, and **O** for oblique (similar to italic). Whether the slant is designated as **I** or **O** depends on the family.

Point Size is the size of the letters. The number refers to decipoints. Multiply the point size to get the decipoints (10 point type is 100 decipoints).

17. Font Control **< previous | content | next >**

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

17. Font Control

Fonts installed on your system can be used by any graphical X Window System application. A remote font server is started automatically with your graphical environment, and then all fonts installed on your system are automatically available to all of your X Window System applications. X clients request character information from the font server, which can return data in various formats, ensuring high quality display of a variety of font styles and sizes.

You can use your familiar host-based X fonts via XLitePro's XFontset Service. The XFontset Service utility provides:

- Rasterized fonts with outline data to all X applications
- Wide range of local X fonts (the standard font sets supplied by MIT along with X11 R6)
- Support for different locale 16-bits fonts including Chinese, Japanese, or Korean fonts, etc.
- Use of uncompressed PCF, SNF, and BDF format files as well as these fonts compressed with the 'compress' or 'gzip' utility
- Font Compiler to compile Microsoft Windows fonts into the X11 format
- Microsoft Windows fonts support in X-sessions.

This chapter describes how you can manage local and remote font sources for XServer to access and to resolve all font requests from X clients.

The **Font Control** tab of **XSetup** in the **Settings** dialog box lets you specify how XServer will use font sources that can be available in X-sessions (including scaled fonts).

The **Pseudo fonts** button in the tab allows you to define how XServer will use MS Windows fonts in X-sessions.

You can run **Font Manager** from the X-session Control menu (or run **XFontset Service** from the XLitePro's Program Folder) to view:
- the ordered font sources list
- the font list of each font directory
- font information, font properties, font metrics
- images of any available font.

If the **Trace Fonts Requests** check box in the **Trace** tab of **XSetup** in the **Settings** dialog box is checked, then all font requests from X clients (with resolve messages) will be stored in the **xserver.out** file. This option is useful for analysis of the font accessibility and resolving font problems by XServer.

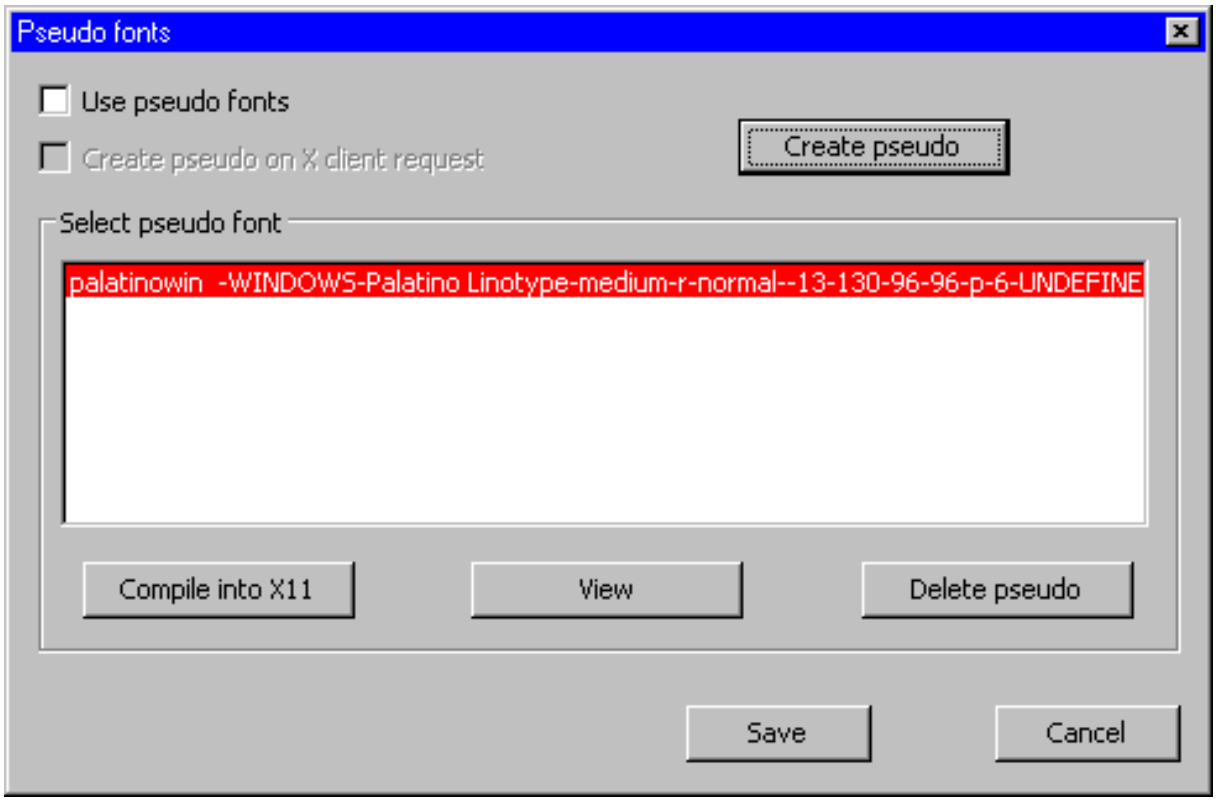
Pseudo Fonts

If you plan to use MS Windows Fonts in the X-session, press the **Pseudo fonts** button in the **Font Control** tab in the **Settings** window. Then you can choose MS Windows fonts and assign aliases to them. This means creating pseudo fonts.

You can compile a pseudo font into the X11 format and save it in any font directory for later use. You can also create a pseudo font directly in the X-session on particular X client's font requests. To enable this, switch on the **Use pseudo fonts** and **Create pseudo on X client request** check boxes in the **Pseudo fonts** dialog box. Pseudo fonts created in the X-session will only become accessible after restarting XServer.

If you need some X font that is inaccessible, XServer allows you to use any MS Windows font instead of it. MS Windows fonts do not support X11 font naming conventions. Therefore, pseudo names (aliases) are used to access them. To use a MS Windows font, you should create a pseudo font for it, i.e. choose one and assign a desirable alias. Pseudo fonts can be created in advance or immediately in the X-session.

You can manage pseudo fonts by using the **Pseudo fonts** dialog box.



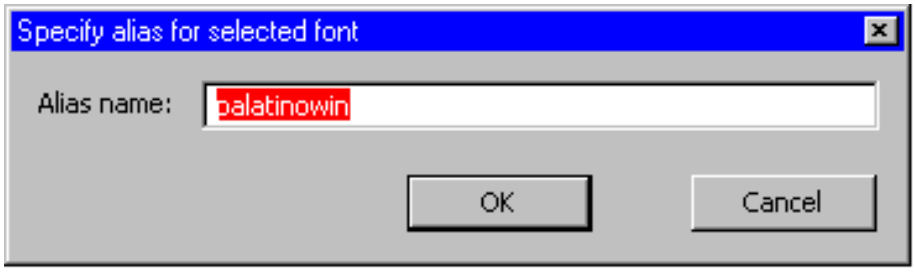
The **Save** button writes all changes to Pseudo Fonts Directory. This cannot be undone if you press **Cancel** in the **Pseudo fonts** dialog box or in the **Font Control Tab** afterwards.

Pseudo Fonts Directory

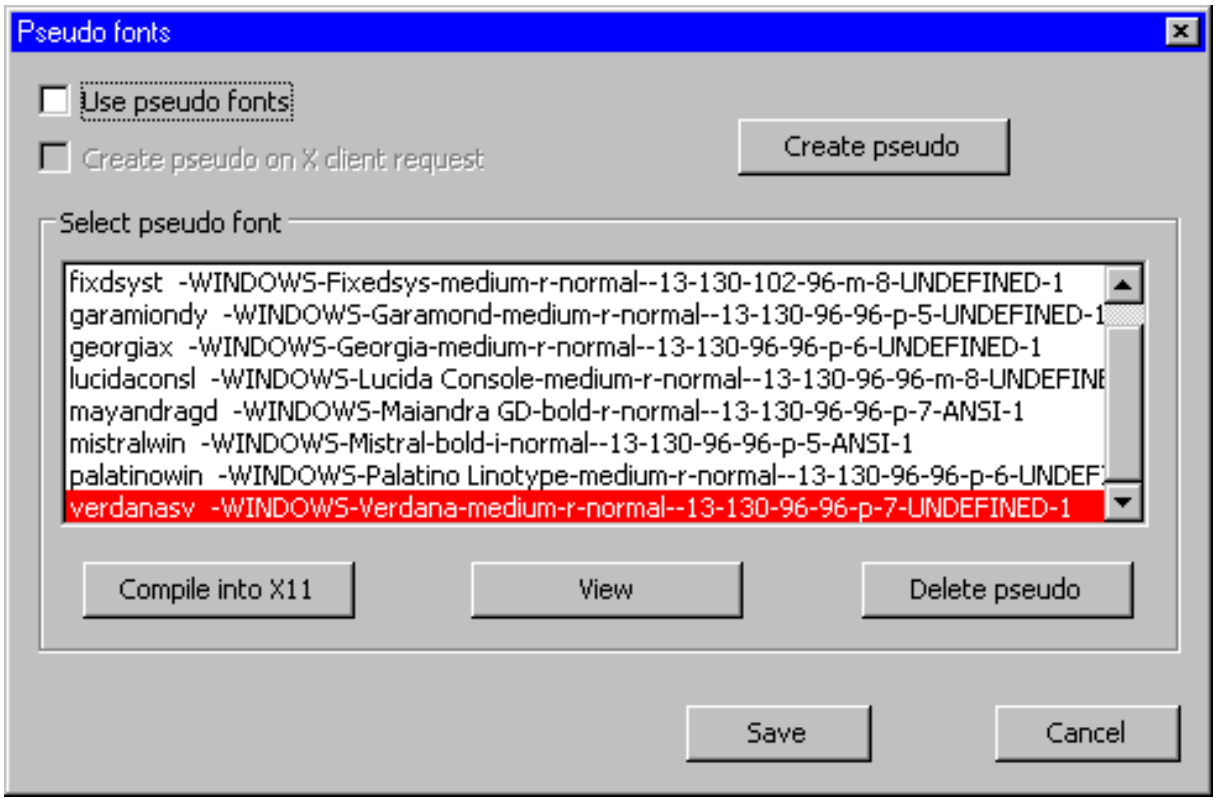
MS Windows font specifications and aliases for Pseudo Fonts are stored in the Pseudo Fonts Directory. The Pseudo Fonts Directory, WINFONTS subdirectory, is located under the home directory (by default) of the software package.

Creating Pseudo Font

To create a pseudo font, press the **Create pseudo** button in the **Pseudo fonts** dialog box. The **Font** dialog box will appear on your display. Select a suitable font and press **OK**. In the **Specify alias for selected font** box, enter a pseudo font alias and click **OK**.



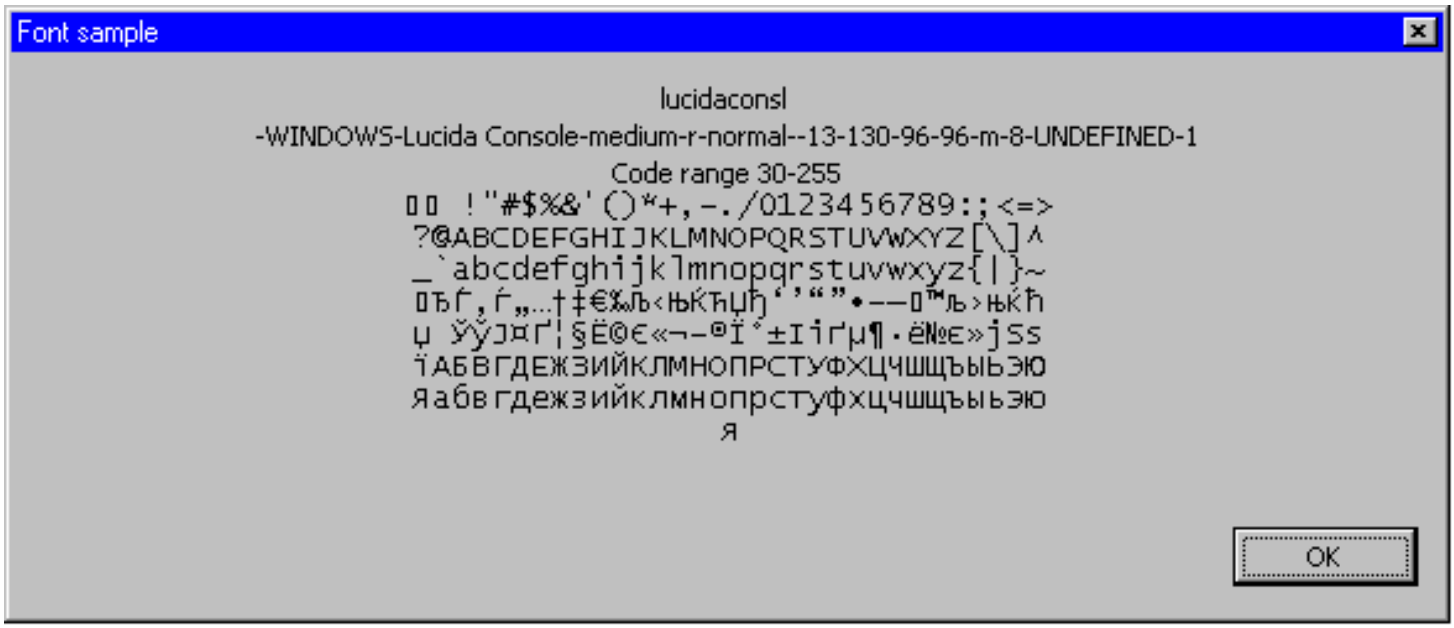
The created pseudo font will appear in the **Select pseudo font** list box lexicographically ordered.



The alias must only contain alphanumeric characters (including the 'underscore' sign). The alias must be unique among existing pseudo fonts.

Viewing Pseudo Font

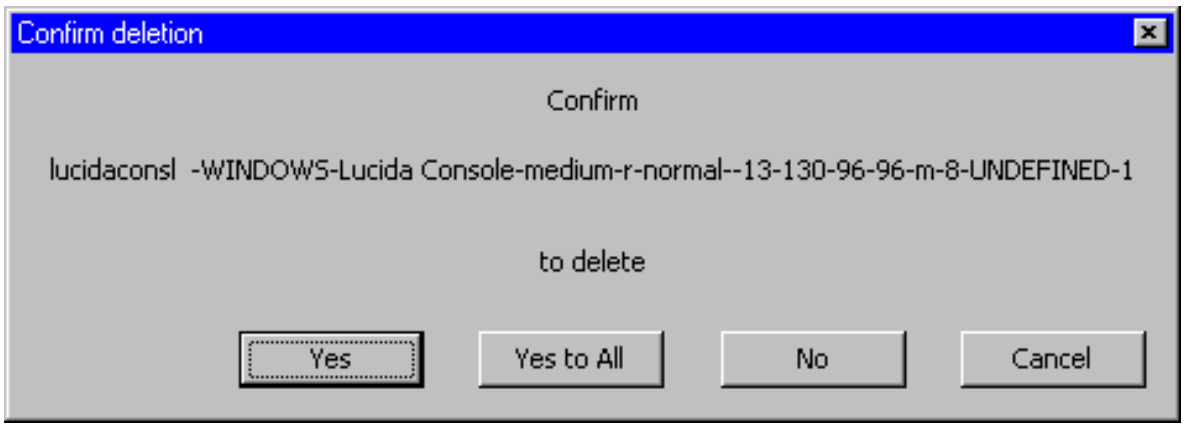
To view a pseudo font, select it in the **Select pseudo font** list box and click the **View** button. The **Font sample** window will appear with the font sample in it.



Note that the font image is displayed by the MS Windows (not by XServer). This option is for font identification only. To see how XServer displays a font image, you can use the '**xfd -fn <alias>**' command in the X-session.

Deleting Pseudo Font

To delete a pseudo font, highlight it in the **Select pseudo font** list box and click the **Delete pseudo** button. You can delete more than one font at a time.

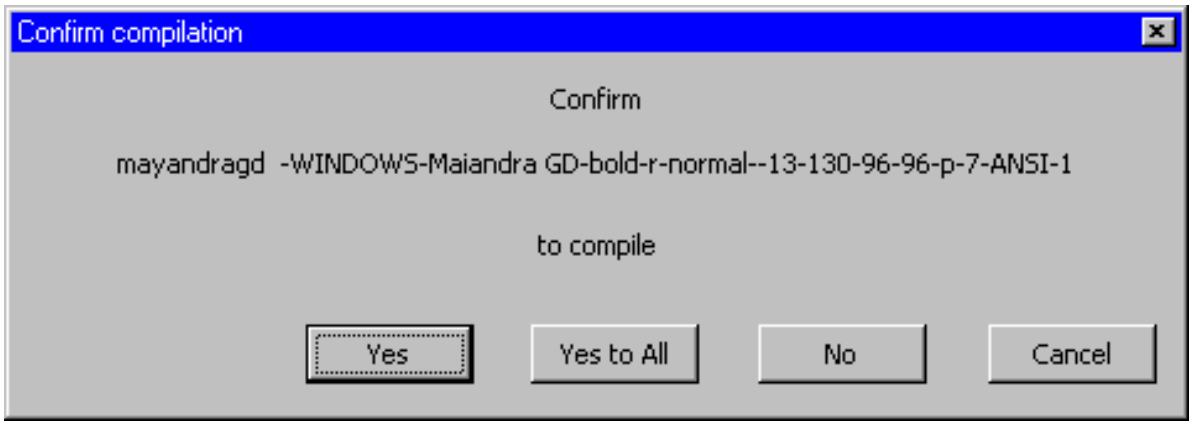


You should confirm removing each font and the total number of the selected fonts.

Compiling Pseudo Font to the X11 format

Loading pseudo fonts takes some time to create their images. Reading font image from the X11 format file is faster. It becomes essential if X client uses a lot of fonts. To prevent this time loss, you can save images of frequently used pseudo fonts in the X11-like format in your Pseudo Fonts Directory. This will require some disk space. To save the image, highlight the required font in the **Select pseudo font** list box and click the **Compile into X11** button. In the **Place X11 font to** box, specify a font directory to store the X11 font file.

You can enter the font directory path in the **Font directory** edit field or use the **Browse** button to select it. If the specified directory is not the font directory (e.g., it is empty), then the **fonts.dir** file will be created in it.



You should confirm compiling each font. To use the font directory, include it into Font Path.

Creating Pseudo Fonts on X Client Request

Pseudo fonts can be created immediately in the X-session on particular requests of X clients. If the requested font is not found, you will be prompted to create a new pseudo font. Creating pseudo fonts in the X-session is especially useful when you run X applications for the first time.

Creating pseudo fonts in an X-session is the same as in the **Pseudo fonts** dialog box. You can use the requested font name as the pseudo font alias. To enable creating pseudo fonts in the X-session, check on the **Create pseudo on X client request** check box. For batch X clients, it is preferable to disable the option to avoid interactive requests.

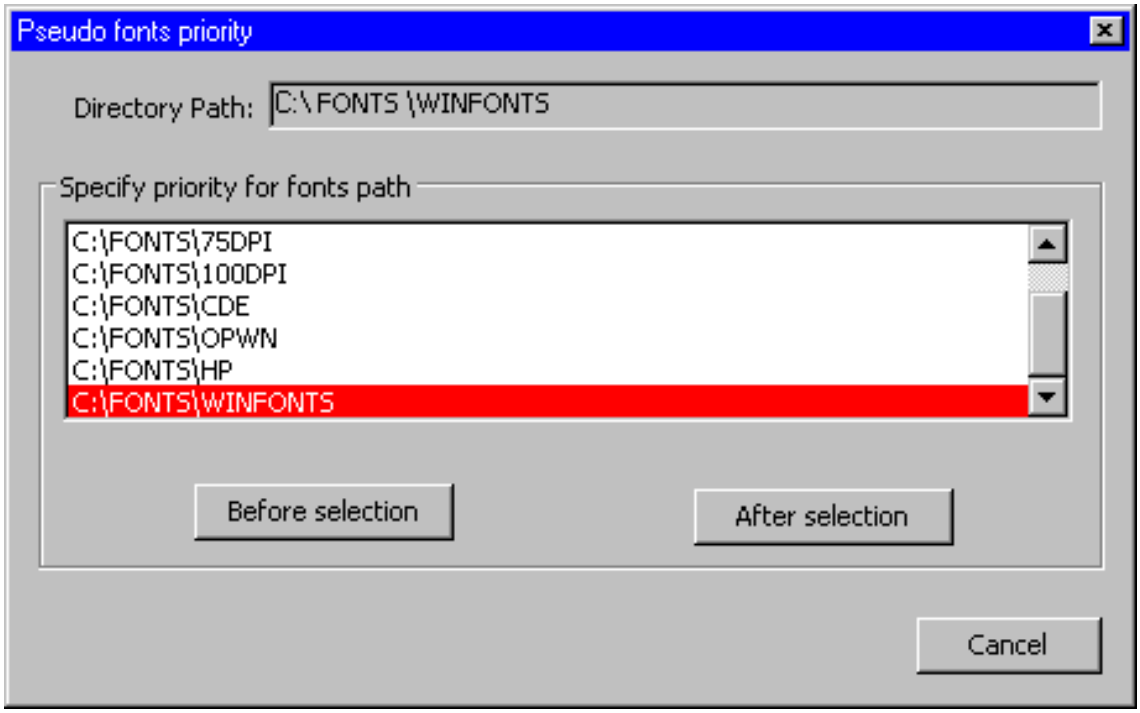
Unfortunately, pseudo fonts created on X client's requests become accessible only after restarting the X-session. In the current X-session, XServer uses the default font instead of requested one.

Enabling to Use

You can define whether to use pseudo fonts in the X-session or not.

To disable pseudo fonts, check off the **Use pseudo fonts** check box. Pseudo Fonts Directory will be removed from Font Path. XServer will only use X fonts. You will not be able to create pseudo fonts in the X-session.

To enable pseudo fonts, check on the **Use pseudo fonts** check box. The **Pseudo fonts priority** dialog box will appear.



You should define priority of Pseudo Fonts Directory among other font sources in Font Path. To insert Pseudo Fonts Directory into Font Path, select a position and use the **Before selection** or **After selection** buttons. XServer will use pseudo fonts as well as X fonts.

17. Font Control **< previous | content | next >**

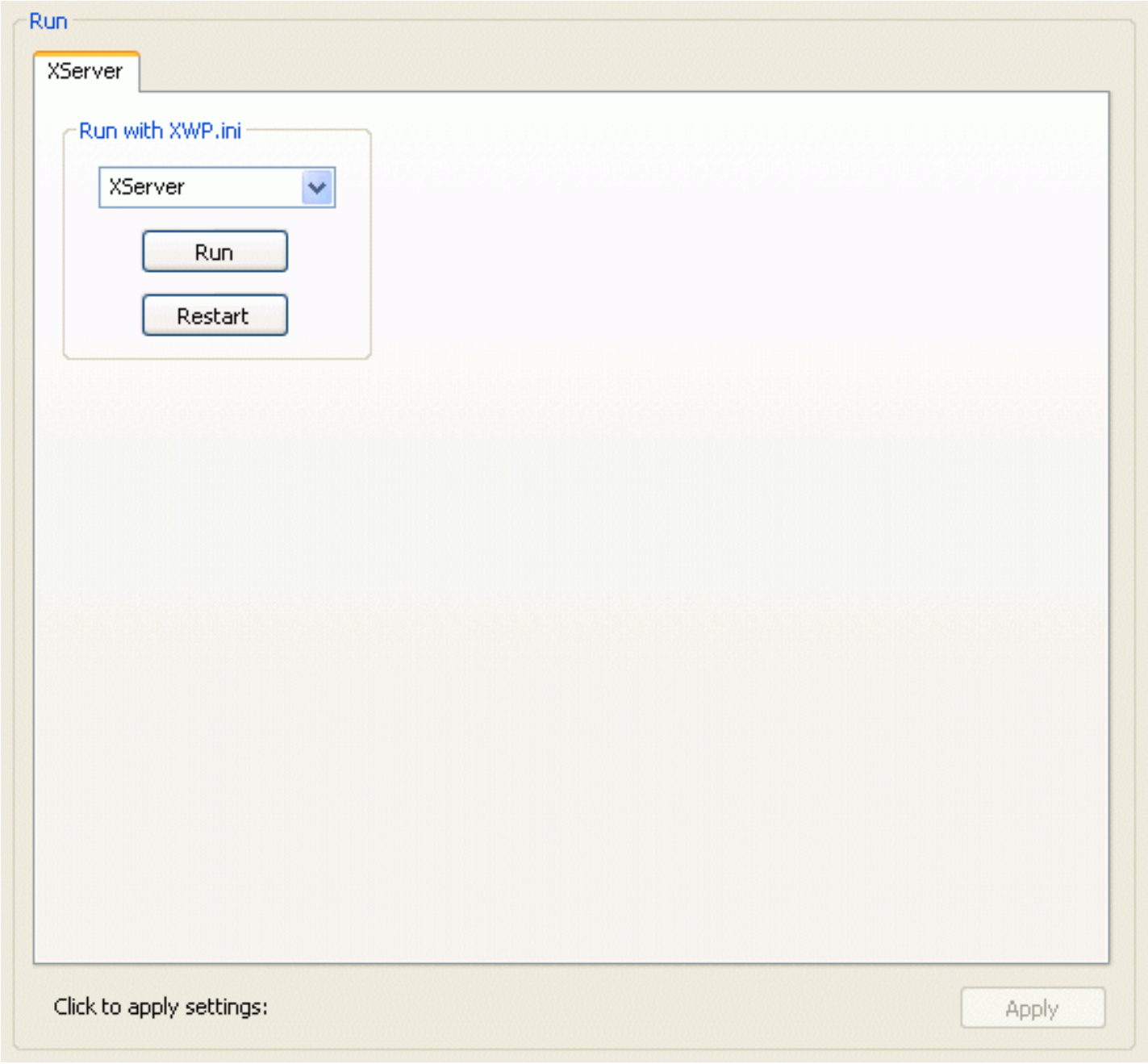
Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

The Run Box

Options that you can select are for running services.

- The XServer Tab

The XServer Tab



The Run with XWP.ini Box

This box allows you to start up or to restart XServer.

Run

This button starts up XServer with current settings if it is not already running (thus beginning a new X-session).

Restart

This button restarts XServer (with reading in the **XWP.ini** file) that is currently being run.

The "[XSETUP]" Section of the ini-file

The "[XSETUP]" section of the ini-file may have the following entry lines you can customize for particular needs and applications. These settings are absent in the XSettings dialog box.

On how to port XServer's settings (i.e., the "[XSETUP]" section of the ini-file) onto other PCs, see subsection **Exporting XServer's Settings During Installation** in section **Running Setup** in chapter **Installing XLitePro**.

AllowCONTEXTMENU=1

You can use this binary setting to enable the standard MS Windows Context Menu (i.e. the standard drop-down menu will appear with right-clicking on a window's title bar).

ConnectXWClipboards=p

This option specifies the mode of connection between MS Windows Clipboard and a current X Selection.

ConnectXWClipboards=7 is the default value (when the **Auto Clipboard Copy&Paste** check box in the XSettings window is checked).

Three of the least significant bits in **p** specify one of three main kinds of auto-copying:

- bit e1=1 enables copying from MS Windows Clipboard to X Selection;
- bit e2=1 enables copying from X Selection to MS Windows Clipboard;
- bit e3=1 makes copying from X Selection to MS Windows Clipboard when any window controlled by XServer loses its activity.

ConnectXWClipboards=0 disables automatic copying MS Windows Clipboard and X Selection when they are changed, i.e. there is no immediate link between MS Windows Clipboard and X Selection (when the **Auto Clipboard Copy&Paste** check box in the XSettings window is not checked).

DisableRENDERAnimCursor=1

You can use this binary setting to display the standard arrow cursor instead of real cursors of X-Clients. This helps accelerate displaying cursors.

ILikeAIX=1

This option specifies the "old-style" keyboard mapping (like it was on the AIX X console, X-Terminals, etc.). This is the actual mode for some AIX X-Clients that work with fixed keyboard mapping only.

ILikeXWinPro52=1

This binary setting allows XServer to only use the "PseudoColor" visual with the "888" colors weight instead of the default value of "666".
This mode of XServer (in the 8-bits mode) is useful for some old CAD/CAE systems.

MiddleButtonTimeout=10

This option is used to set up the value for the 'MiddleButton wait timeout', in msec.

TerminateOnSuspend=1

You can use this binary setting to provide a graceful termination of X-Clients on suspending PCs.

Emulate WM window substrate=1

This option is for the **Multiple** mode of XServer.
In this mode, the local window manager provides usual behavior, but each top-level non-popup X window has a parent window controlled by the window manager.

UnlatchedCapsLock=1

This option toggles the actual CapsLock keyboard key to unlatched mode, i.e. XServer will consider the CapsLock key as a normal key (non-toggling).

This is the actual setting if you re-map the CapsLock key in the kmf-file (say, interchanging the CapsLock and LeftControl keys, that is popular among some users on SUN machines).

Use_8_3_fonts_alias_file=1

This option specifies for XServer to use the standard **fonts.alias** file name.

Historically, XServer uses the **fonts.ali** name to list font aliases in the file.

This setting is useful when XServer uses a mounted font directory.

UseUsualSelect=1

This setting provides the "old-style" way of using the **select** WinSock API request.

Restriction of Windows Fonts Accessibility

The current version of XServer provides access only to those MS Windows fonts that are available in the MS Windows System Font Table. Therefore, accessibility of a particular MS Windows font depends upon MS Windows session history. If you try to use a MS Windows font that is inaccessible, MS Windows will give you some other font with the closest metrics. XServer guarantees getting of the exactly requested MS Windows font if you create the pseudo font for it immediately after starting MS Windows or compile it into the X11 format file.

7. Telnet

This chapter describes how to start and use the Telnet program supplied with XLitePro.

Telnet is a communications and terminal emulation program for logging into remote machine and executing commands in a remote machine. It allows you to connect to and communicate with hosts that support the Telnet protocol and run a Telnet service over an insecure channel.

To provide terminal emulation from your computer running MS Windows, the remote host must be configured with the TCP/IP program, the Telnet service program (daemon), and a user account for your PC.

Once you have established a connection, you can use Telnet to start X clients and perform other operations outside the X Window System environment.

While you are using Telnet, your PC emulates one of the following terminal types: XTERM, ANSI, AT386; DEC VT52, VT100, VT125, VT220 or VT240, using connection-based services of TCP. You can specify the terminal emulation settings for the current connection by making the appropriate settings on the Settings option.

By using the **Keyboard Mapping** option (i.e. keymap editor invoking), you can load, change (re-define keys and create a new keyboard layout), and save any keyboard definition file.

You can start more than one Telnet session, and use Telnet to open multiple Telnet windows on a single host or different hosts at the same time.

9. Using XServer

The XLitePro's XServer is a program that emulates the X terminal on your PC. It is the X-server implementation of the X11R6.6 release of the X Window System. XServer allows you to run one or more X Window based client applications (X clients) that are resident on host computers. The host can be any computer that supports the X protocol across your TCP/IP network.

X client applications can be displayed in individual windows, as multiple windows contained in a single XServer's window, or in a full-screen mode outside the Microsoft Windows graphical environment. The first two methods include functions to copy and paste data between X clients and the Microsoft Windows clipboard.

While running XServer, you can initiate a login session by using Telnet, XDMCP, or Startup (REXEC, RSH, RLOGIN) methods. You can launch several X-sessions that will be running simultaneously each with its own settings. XServer can work on multi-monitors PCs.

While running XServer, you can launch any local X client supplied with the package.

The macro options of XServer let you create macro-files to remember sequences of your keystrokes. Then, whenever you want, you can play back these keystroke macros in a target X-application window by choosing an appropriate macro-file.

XServer provides German and/or French messages and text labels under the corresponding national version of MS Windows.

XServer supports the following X Extensions:

- Extended-Visual-Information
- XinputExtension
- MIT-SUNDRY-NONSTANDARD
- SHAPE
- XTEST
- XC-MISC
- XC-APPGROUP
- GLX for OpenGL
- RENDER
- SECURITY
- LBX

To get X Extensions supported by your system, run the following command line:

xdpyinfo -display IP_address_of_your_PC:0

XServer can connect to a remote LbxLoxy service (a Low Band Width X proxy) either by direct access to a remote host or through one of the established SSH1/SSH2 protocol connections as well.

XServer can use MESA (6.2.1), an open-source implementation of the OpenGL specification.

This chapter describes how to use XServer in all four possible modes: **Full Screen** mode, **Single** window mode, **Multiple** window mode, and **Multiple+RemoteWM** mode.

< previous | content | next >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

Using LBX

About LBX

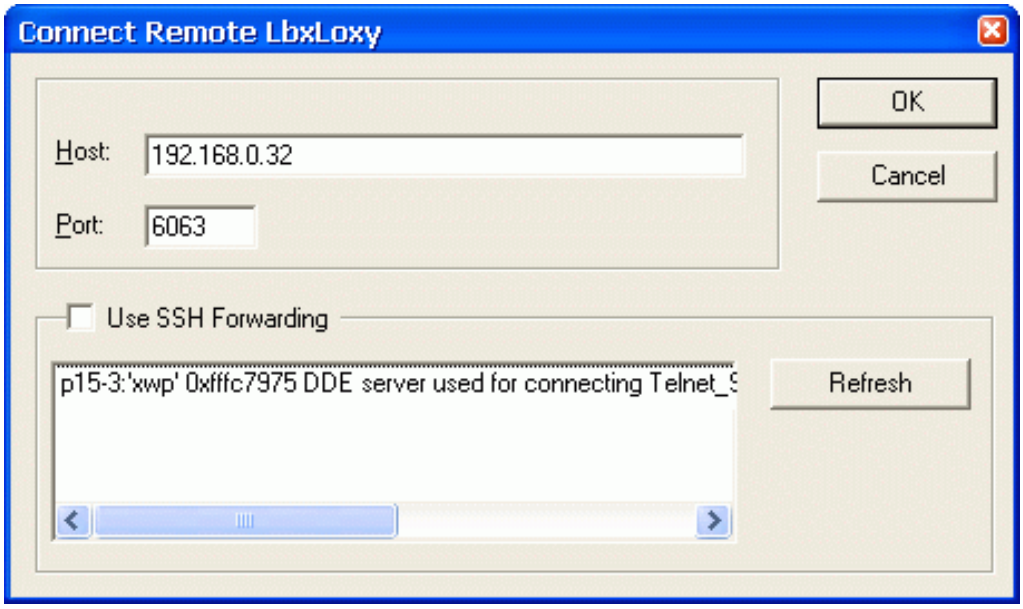
Low Bandwidth X (LBX) is a network-transparent protocol for running X Window System applications over transport channels whose bandwidth and latency are significantly worse than that used in local area net works.

The design centre for LBX is to use a proxy as an intermediary between the client and server, so that the low bandwidth/high latency communication occurs between the proxy and server. The proxy re-encodes and compresses requests, events, replies and errors, as well as the resulting data stream to reduce the volume of data that must be sent over the wire. Additionally, the proxy can cache information from the server to provide low-latency replies to clients. This reply generation by the proxy is known as short-circuiting. A proxy can handle multiple clients for a given server, but does not prevent clients from connecting directly to the server. The design allows the proxy to multiplex multiple clients into a single data stream to the server.

LBX employs several different compression and short-circuiting methods. Use of these methods is negotiable, and in some cases, the algorithm used by a given method is negotiable as well. LBX also provides for negotiation of extensions to LBX.

About lbxproxy

Applications that would like to take advantage of the Low Bandwidth extension to X (LBX) must make their connections to an lbxproxy. These applications need to know nothing about LBX, they simply connect to the lbxproxy as if it were a regular server. The lbxproxy accepts client connections, multiplexes them over a single connection to the X server, and performs various optimizations on the X protocol to make it faster over low bandwidth and/or high latency connections.



With regard to authentication/authorization, lbxproxy simply passes along to the server the credentials presented by the client. Since X clients will connect to lbxproxy, it is important that the user's **.Xauthority** file contain entries with valid keys associated with the network ID of the proxy. lbxproxy does not get involved with how these entries are added to the **.Xauthority** file. The user is responsible for setting it up.

Synopsis of lbxproxy:

lbxproxy [:<display>] [option]

If **:<display>** is specified, the proxy will use the given display port when listening for connections. The display port is an offset from port 6000, identical to the way in which regular X display connections are specified. If no port is specified on the command line option, lbxproxy will default to port 63. If the port that the proxy tries to listen on is in use, the proxy will exit with an error message.

The lbxproxy program has various options, all of which are optional. It is recommended to use them with their defaults.

The **-display dpy** option specifies the address of the X server supporting the LBX extension. If this option is not specified, the display is obtained by the DISPLAY environment variable.

See lbxproxy manual for more details.

Running xterm from Telnet

Suppose that your PC has the network name xtp. Run XServer, then run Telnet and connect to remote machine supporting LBX. After logging in, you can launch lbxproxy and then xterm in the main Telnet window by using the following commands (**$** is a prompt sign):

```
$ lbxproxy  :62  -display xtp:0 &
$ xterm  -display :62 &
```

or (with setting the DISPLAY environment variable and using it by default):

```
$ DISPLAY= xtp:0;  export  DISPLAY
$ lbxproxy  :62 &
$ xterm  -display :62 &
```

After that xterm will work with display :62.0 (corresponding to port 6062) and will use LBX protocol. In the xterm window, you can run the **echo $DISPLAY** command resulting in :62.0 (for our example).

If the port that the proxy tries to listen on (6062) is in use, the proxy will exit with an error message. To locate available proxy services, you can use the xfindproxy program communicating with a proxy manager (see **About xfindproxy** below).

Terminating XSession terminates lbxproxy.

You can launch more than one lbxproxy-xterm pairs, each with different displays available (e.g., :61.0 in addition to above).

Note: to run lbxproxy under Solaris, you must have rights as root.

Running xterm from Startup

The Startup utility automatically runs XSession and then prompts you to specify data for connection and commands for execution. You can connect to remote machine supporting LBX and start up lbxproxy-xterm pair by entering in the Command field and sequentially executing the following commands (e.g., from xtp:0 and for port 6061):

```
/usr/X11R6/bin/lbxproxy  :61  -display xtp:0
/usr/X11R6/bin/xterm  -display :61
```

Note that you should enter the full file specification to execute such commands. (You can use **locate lbxproxy** or **which lbxproxy** commands to determine the path on the remote machine.)

Note that **-zlevel 6** is the most optimal option you can set for the Zlib compression level used by lbxproxy for stream compression.

About xfindproxy

xfindproxy is a program used to locate available proxy services. It utilizes the Proxy Management Protocol to communicate with a proxy manager. The proxy manager keeps track of all available proxy services, starts new proxies when necessary, and makes sure that proxies are shared whenever possible (see **About proxymngr** below).

Synopsis of the xfindproxy:

xfindproxy -manager managerAddr -name serviceName -server serverAddr [-auth] [-host hostAddr] [-options opts]

The **-manager** argument is required, and it specifies the network address of the proxy manager. The format of the address is a standard ICE network id (e.g. "tcp/a.b.c:6500").

The **-name** argument is required, and it specifies the name for the desired proxy service (e.g. "LBX"). The name is case insensitive.

The **-server** argument is also required, and it specifies the address of the target server. The format of the address is specific to the proxy service specified with the -name argument. For example, for a proxy service of "LBX", the address would be an X display address (e.g. "xtp:0").

The **-auth** argument is optional. If specified, xfindproxy will read 2 lines from standard input. The first line is an authorization/authentication name. The second line is the authorization/authentication data in hex format (the same format used by xauth). xfindproxy will pass this auth data to the proxy, and in most cases, will be used by the proxy to authorize/authenticate itself to the target server.

The **-host** argument is optional. If xfindproxy starts a new proxy service, it will pass the host specified. The proxy may choose to restrict all connections to this host. In the event that xfindproxy locates an already existing proxy, the host will be passed, but the semantics of how the proxy uses this host are undefined.

The **-options** argument is optional. If xfindproxy starts a new proxy service, it will pass any options specified. The semantics of the options are specific to each proxy server. In the event that xfindproxy locates an already existing proxy, the options will be passed, but the semantics of how the proxy uses these options are undefined.

If xfindproxy is successful in obtaining a proxy address, it will print it to stdout. The format of the proxy address is specific to the proxy service being used. For example, for a proxy service of "LBX", the proxy address would be the X display address of the proxy (e.g. "a.b.c:62").

If xfindproxy is unsuccessful in obtaining a proxy address, it will print an error to stderr.

See xfindproxy manual for more details.

Example

Suppose that your PC has the network name xtp (with XServer running), and the network address of the proxy is "a.b.c:6500". Execute the following command (in Telnet or Startup as described above):

xfindproxy -manager a.b.c:6500 -name lbx -server xtp:0

For a proxy service of "LBX", the proxy address would be the X display address of the proxy (e.g., "a.b.c:62") printed to stdout. (See also **Proxy manager details** below.)

After that you can execute the following command (see **Running xterm from Telnet** above):

```
xterm  -display :62 &
```

Note that XSession must be running in your PC when you execute proxymngr and xfindproxy (xtp:0 in commands above is the address of the target X server).

Note: terminating XSession terminates lbxproxy.

About proxymngr

The proxy manager (proxymngr) is responsible for resolving requests from xfindproxy (and other similar clients), starting new proxies when appropriate, and keeping track of all of the available proxy services. The proxy manager strives to reuse existing proxies whenever possible.

There are two types of proxies that the proxy manager deals with, "managed" and "unmanaged" proxies. A managed proxy is a proxy that is started "on demand'' by the proxy manager. An unmanaged proxy, on the other hand, is started either at system boot time, or manually by a system administrator. The proxy manager is made aware of its existence, but no attempt is made by the proxy manager to start unmanaged proxies.

Synopsis of proxymngr:

```
proxymngr  [-config filename] [-timeout seconds] [-retries #] [-verbose]
```

The command line options that can be specified to proxymngr are:

-config

Used to override the default proxymngr config-file. See **Proxy Manager Config-File** below for more details about the config-file.

-timeout

Sets the number of seconds between attempts made by the proxy manager to find an unmanaged proxy. The default is 10.

-retries

Sets the maximum number of retries made by the proxy manager to find an unmanaged proxy. The default is 3.

-verbose

Causes various debugging and tracing records to be displayed as requests are received and proxies are started.

Proxy Manager Config-File

The proxy manager maintains a local configuration file describing the proxy services available. This configuration file is installed in **/usr/X11R6/lib/X11/proxymngr/pmconfig** during the installation of proxymngr. The location of the configuration file can be overwritten using the **-config** command line option.

Aside from lines starting with an exclamation point for comments, each line of the configuration file describes either an unmanaged or managed proxy service.

For unmanaged proxies, the format is:

```
<service-name> unmanaged <proxy-address>
```

service-name

Is the name of the unmanaged proxy service, and must not contain any spaces, for example "XFWP''. **service-name** is case insensitive.

proxy-address

Is the network address of the unmanaged proxy. The format of the address is specific to the service-name. For example, for the "XFWP'' service, the proxy-address might be "firewall.x.org:100''.

If there is more than one entry in the config-file with the same unmanaged service-name, the proxy manager will try to use the proxies in the order presented in the config-file.

Example

xfwp unmanaged firewall:4444

For managed proxies, the format is:

<service-name> managed <command-to-start-proxy>

service-name

Is the name of the managed proxy service, and must not contain any spaces, for example "LBX''. **service-name** is case insensitive.

command-to-start-proxy

Is the command executed by the proxy manager to start a new instance of the proxy. If **command-to-start-proxy** contains spaces, the complete command should be surrounded by single quotes. If desired, **command-to-start-proxy** can be used to start a proxy on a remote machine.

Example

lbx managed /usr/X11R6/bin/lbxproxy

Proxy Manager Details

When the proxy manager gets a request from xfindproxy (or another similar client), its course of action will depend on the service-name in question.

For a managed proxy service, the proxy manager will find out if any of the already running proxies for this service can handle a new request. If not, the proxy manager will attempt to start up a new instance of the proxy (using the **command-to-start-proxy** found in the config-file). If that fails, an error will be returned to the caller.

For an unmanaged proxy service, the proxy manager will look in the config-file to find all unmanaged proxies for this service. If there is more than one entry in the config-file with the same unmanaged service-name, the proxy manager will try to use the proxies in the order presented in the config-file. If none of the unmanaged proxies can satisfy the request, the proxy manager will timeout for a configurable amount of time (specified by **-timeout** or default of 10) and reattempt to find an unmanaged proxy willing to satisfy the request. The number of retries can be specified by the **-retries** argument, or a default of 3 will be used. If the retries fail, the proxy manager has no choice but to return an error to the caller (since the proxy manager cannot start unmanaged proxy services).

See proxymngr manual for more details.

To run proxymngr, check its config-file and execute the following command with options you need:

/usr/X11R6/bin/proxymngr

Note: to run proxymngr, you must have rights as root.

Appendix A Keyboard Mapping File Format

Keyboard Mapping File Format

The Keyboard Mapping file has two sections, [KEYS] and [COMPOSERS_XKK], each consisting of keysym statements and possible comments. Text entered after a ';' sign is ignored and will be treated as a comment.

A keysym specification uses a set of standard X numbers to describe a symbol. For example, a lowercase 'a' has a special number code.

The KEYS Section

In the [KEYS] section, each keysym statement associates a set of one to four keysyms with a physical key.

Synopsis of a statement is:

KEYnn = keysym1 [, keysym2 [, keysym3 [, keysym4]]]

'nn' is the identifying number of a PC's key. Every PC's key has an entry in the section. The entry name is the text **KEY** followed by the decimal ScanCode number (key_number) and followed by the letter 'E' if the key has the extended flag set.

keysym1 is the keysym associated with the key in a non-shifted state (Normal). This is the only parameter that must be entered.

keysym2 is the keysym associated with the key when the key is Shifted.

keysym3 is the keysym associated with the key when a **Mode-Shift** key is pressed (Alt-Gr).

keysym4 is the keysym associated with the **Shift** + **Mode-Shift** sequence (Shift + Alt-Gr).

A **Mode-Shift** is a physical key which has a keysym value of 0xFF7E, (predefined as the ModeSwitch symbol), and which is assigned to one of the modifiers MOD1 to MOD5.

A full keysym specification consists of four numbers, each of which is in the range of 0 through 255 decimal (or 0x00 through 0xFF hex). The standard predefined X keysyms use only the third and fourth numbers. The first two numbers are assumed to be zero.

XLitePro accepts keysyms in the following three formats:

1. In the dotted notation, where up to four numeric components are separated by periods '.'.

Each numeric component represents one of the four numbers that define a keysym. If a component is omitted, it is assumed to be zero in the left-most position. If two components are omitted, the two left-most components are assumed to be zero, etc. For example, if you enter the numbers

32.255

two of the four possible components are omitted. The keysym will be interpreted as

0.0.32.255

In the dotted notation, the two lines below both represent the same keysym:

255.0xFE

0.0.255.0xFE

2. A single numeric value containing up to four bytes specifications. Unspecified numbers are assumed to be zeroes in the left-most position. For example:

   0xFF20

   represents the values 0xFF and 0x20. The keysym is interpreted as follows:

   0.0.0xFF.0x20

   Predefined symbols can be used instead of the keysym formats described above. The following three symbols are predefined:

   ModeSwitch0xFF7E

   VoidSymbol0xFFFFFF

   NoSymbol0

   You can get a full list of X-keys via the **xmodmap** utility of UNIX by using the **-pm** and/or **-pk** options.

   The values of keysyms for keys may be obtained via the **/usr/openwin/demo/xev** X-Window's UNIX utility.

3. A keysym value can be of a composer type, i.e. the **COMPxx** entry exists in the [COMPOSERS_XKK] section with the keysym value 'xx'. See the [COMPOSER_XKK] section below for details of how composers work.

Examples:

```
[KEYS]
KEY30 = 97, 65; LATIN LETTER a / A
; KEY30 = 0x61, 0x41; (XK_a, XK_A); just the same as the previous line
KEY80E= 255.84, 255.84; cursor DOWN / cursor DOWN
; KEY80E=0xff54, 0xff54; (XK_Down, XK_Down); just the same
```

Examples of re-mapping

Suppose that you selected to use the **us.kmf** file (with XSettings) and you want to change the keyboard mapping for Up(8) and Down(2) keys of the additional keyboard and for the F1 key. You can find the following lines in the **us.kmf** file for them:

```
KEY59 = 0xffbe, 0xffbe ; (XK_F1,XK_F1)
.........
KEY72E = 0xff52, 0xff52 ; (XK_Up,XK_Up)
KEY80E = 0xff54, 0xff54 ; (XK_Down,XK_Down)
```

By using the **xev** or **xmodmap -pk** commands on the SUN host, you can get keysyms for all keys. For example, you choose:

```
Help=0xff6a;
Paste=0xffcf;
Copy=0xffcd;
```

If you modify the lines of the **us.kmf** file as follows:

```
KEY59 = 0xff6a, 0xff6a ; (Help,Help)
.........
KEY72E = 0xffcf, 0xffcf ; (Paste,Paste)
KEY80E = 0xffcd, 0xffcd ; (Copy,Copy)
```

you will emulate the "Help" (F1), "Copy" ("Down") and "Paste" ("Up") functions of the SUN keyboard.

Another example of keyboard mapping

Suppose that you use the **us.kmf** file and you want some keys of your keyboard to map to the SUN one. You can find the following lines in the **us.kmf** file:

```
KEY59 = 0xffbe, 0xffbe ; (XK_F1,XK_F1)
KEY60 = 0xffbf, 0xffbf ; (XK_F2,XK_F2)
KEY61 = 0xffc0, 0xffc0 ; (XK_F3,XK_F3)
KEY62 = 0xffc1, 0xffc1 ; (XK_F4,XK_F4)
KEY63 = 0xffc2, 0xffc2 ; (XK_F5,XK_F5)
KEY64 = 0xffc3, 0xffc3 ; (XK_F6,XK_F6)
KEY65 = 0xffc4, 0xffc4 ; (XK_F7,XK_F7)
KEY66 = 0xffc5, 0xffc5 ; (XK_F8,XK_F8)
KEY67 = 0xffc6, 0xffc6 ; (XK_F9,XK_F9)
KEY68 = 0xffc7, 0xffc7 ; (XK_F10,XK_F10)
```

If you change them to:

```
KEY59 = 0xffc8, 0xffc8, 0xffbe ; (XK_F11,XK_F11,XK_F1)
KEY60 = 0xffc9, 0xffc9, 0xffbf ; (XK_F12,XK_F12,XK_F2)
KEY61 = 0xffca, 0xffca, 0xffc0 ; (XK_F13,XK_F13,XK_F3)
KEY62 = 0xffcb, 0xffcb, 0xffc1 ; (XK_F14,XK_F14,XK_F4)
KEY63 = 0xffcc, 0xffcc, 0xffc2 ; (XK_F15,XK_F15,XK_F5)
KEY64 = 0xffcd, 0xffcd, 0xffc3 ; (XK_F16,XK_F16,XK_F6)
KEY65 = 0xffce, 0xffce, 0xffc4 ; (XK_F17,XK_F17,XK_F7)
```

KEY66 = 0xffcf, 0xffcf, 0xffc5 ; (XK_F18,XK_F18,XK_F8)
KEY67 = 0xffd0, 0xffd0, 0xffc6 ; (XK_F19,XK_F19,XK_F9)
KEY68 = 0xffd1, 0xffd1, 0xffc7 ; (XK_F20,XK_F20,XK_F10)

then the following keys of your keyboard will map to keys of the SUN keyboard:

F1 => F11/Stop
F2 => F12/Again
F3 => F13/Props
F4 => F14/Undo
F5 => F15/Font
F6 => F16/Copy
F7 => F17/Open
F8 => F18/Paste
F9 => F19/Find
F10 => F20/Cut

To get codes of F1 - F10 keys, you should press them together with the Alt-Gr key.

The COMPOSERS_XKK Section

In many European languages (especially in France, Belgium and Holland), users need to enter some special characters by combining a Diacritic (or composer) character and a normal letter. For example, the user enters first the '^' sign and then the 'a' character, then this should result in the 'b' keysym.

The [KEYS] section does not determine composer characters. The Composers are only defined in the [COMPOSERS_XKK] section.

In the [COMPOSERS_XKK] section, each composer statement associates a set of key_ number map pairings with a keysym value.

Synopsis of the composer statement is:

COMPxx = key_number > key_number[S] [, key_number > key_number[S] ...]

In the composer entry, **COMP** is the entry name and 'xx' is a decimal keysym value for a composer key (in the range of character codes). The '>' sign defines single code mapping (from the left to the right), while a comma separates possible map pairings. The 'S' character, if exists, allows both cases for a key_number mapping pair, otherwise lower case only.

If for a keysym value 'xx' of a key (say, KEYcc), a composer entry COMPxx exists in the [COMPOSERS_XKK] section (i.e. XServer can find it there), then the 'cc' value will not be sent to the X Client (otherwise, it will).

In the composer case, XServer will save the keysym value 'xx' until the user presses the next key. If the next key (say, KEYyy) is in the COMPxx entry (like 'yy > zz' in a pair), then XServer will send the value 'zz' from the pair to the X Client. If 'yy' is not found in the COMPxx entry, then XServer will send the composer's key_number 'cc' and the second key_number 'yy'.

Note that the values 'yy' and 'zz' are in the range of character codes.

Note that if Composer is pressed twice, then XServer will send the single value 'cc' to the X Client.

Example:

```
[KEYS]
KEY18 = 0x65, 0x45; (XK_e, XK_E)
KEY22 = 0x75, 0x55; (XK_u, XK_U)
KEY23 = 0x69, 0x49; (XK_i, XK_I)
KEY24 = 0x6f, 0x4f; (XK_o, XK_O)
KEY30 = 0x61, 0x41; (XK_a, XK_A)
KEY41 = 94, 176; Circumflex Accent (^) / DEGREE SIGN, RING ABOVE
; KEY41 = 0x5e, 0xb0; (XK_asciicircum, XK_degree)
KEY162 = 0xe2, 0xc2; (XK_acircumflex, XK_Acircumflex)
KEY170 = 0xea, 0xca; (XK_ecircumflex, XK_Ecircumflex)
KEY174 = 0xee, 0xce; (XK_icircumflex, XK_Icircumflex)
KEY180 = 0xf4, 0xd4; (XK_ocircumflex, XK_Ocircumflex)
KEY187 = 0xfb, 0xdb; (XK_ucircumflex, XK_Ucircumflex)

[COMPOSERS_XKK]
; Definition of Circumflex Accent as a composer
COMP94=30>162S,18>170S,23>174S,24>180S,22>187S
```

In this example, if the user presses (normally) the key 41 which is the circumflex accent (on German keyboard), XServer will check if the keysym value 94 is found in the [COMPOSERS_XKK] section (the COMP94 entry), and if yes, then XServer will wait until the user enters the next character. If the next character is in the COMP94 entry (in our case 30), then XServer will send 162 to the X Client (both cases are allowed). If the second key_number is not found in the COMP94 entry, then XServer will send the composer's key_number (in our case 41) and the second key_number.

Two examples of the Scroll_Lock key implementation

The "Scroll_Lock" key is used to minimize the XServer window running in the FullScreen mode (if the Local ScrollLock key check box on the XSettings window is on). But some of keyboards do not have certain keys (e.g., Scroll_Lock). By the example of two files, **us_ps.kmf** and **us_ps2.kmf**, two possible solutions are considered below to implement the "Scroll_Lock" key code generation.

Note that both the files were produced from our standard **us.kmf** file. You can compare them to view the changes made. Of course, these are only EXAMPLES. The similar changes (of KEY60) may be made with ANY existing keyboard key.

- The **us_ps.kmf** file defines the "Escape" key as the COMPOSER key. So, the "Esc"+"i" key sequence will generate the Scroll_Lock key code. In this case, to generate the "simple Escape" key code, you must press the "Esc" key twice.
- The **us_ps2.kmf** file gives another solution - the "Shift"+"F2" key sequence will generate the Scroll_Lock key code.

Available Keyboard Mapping Files

Keyboard Filename	Keyboard
• belgian.kmf	Belgian Keyboard (for 102 Keyboard)
• croatian.kmf	Croatian Keyboard (for 102 Keyboard)

• danish.kmf	Danish Keyboard (for 102 Keyboard)
• decemfr.kmf	DEC style Keyboard Mapping for a French Keyboard
• decemfrc.kmf	DEC style Keyboard Mapping for a French Canadian Keyboard
• decemgr.kmf	DEC style Keyboard Mapping for a German Keyboard
• decemuk.kmf	DEC style Keyboard Mapping for a U.K. 102 English Keyboard
• decemus.kmf	DEC style Keyboard Mapping for a U.S. Keyboard
• dutch.kmf	Dutch Keyboard (for 102 Keyboard)
• dvorak.kmf	Dvorak Keyboard (for 102 Keyboard)
• finnish.kmf	Finnish Keyboard (for 102 Keyboard)
• frencan.kmf	French Canadian Keyboard
• french.kmf	French Keyboard
• german.kmf	German Keyboard
• hungarn.kmf	Hungarian Keyboard
• italian.kmf	Italian Keyboard
• jpn106.kmf	Japanese 106 Keyboard
• latinam.kmf	Latin American Keyboard
• msus.kmf	Microsoft U.S. English Keyboard
• norwegia.kmf	Norwegian Keyboard
• polish.kmf	Polish Keyboard (for 102 Keyboard)
• portugue.kmf	Portuguese Keyboard
• portbraz.kmf	Portuguese (Brazilian) Keyboard
• rus.kmf	Russian Keyboard (for 102 Keyboard)
• rusalt.kmf	Russian Alternative Keyboard (for 102 Keyboard)
• slovenia.kmf	Slovenian Keyboard (for 102 Keyboard)
• spanish.kmf	Spanish Keyboard
• swedish.kmf	Swedish Keyboard
• swissfre.kmf	Swiss French Keyboard
• swissger.kmf	Swiss German Keyboard
• uk101.kmf	U.K. 101 English Keyboard
• uk102.kmf	U.K. 102 English Keyboard
• uk102m.kmf	U.K. 102 English Keyboard (for SWISSGER 102 Keyboard)
• us.kmf	U.S. English Keyboard
• userkbd.kmf	User-defined keyboard mapping

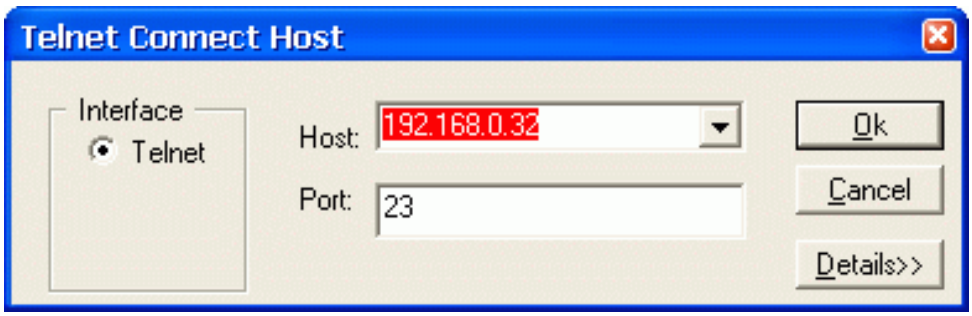
< previous | content | next >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

Starting and Terminating Telnet

You can start Telnet by double-clicking on the **Telnet** icon in the XLitePro Programs' folder:

The **Telnet Connect Host** dialog box will appear on your display:



A session is a connection to a remote machine made with a number of connection-specific settings assigned to it. These settings are saved in an ini-file and profiles and allow you to have different preferences for different hosts (using different ini-files and profiles).

All Telnet session settings are stored in the [TELNET] section of the **xwp.ini** file.

Also you can specify some initial settings for the session with the **Details** button (that is described in section **Details of a Session**).

The first thing you should do to initiate a session is to establish a connection to a remote machine. In the dialog, you must specify the hostname or IP address and the port number of the service.

Pressing **OK** will store current settings (for the next session) and will establish a connection, using them.

You can cancel any changes you have made to the dialog box by clicking on **Cancel**. This will also close the dialog.

Telnet

This button specifies whether to use the standard TELNET protocol over an insecure channel to provide an interface for communications between clients and servers. In the **Telnet** mode mode, Telnet works as the Telnet client on your PC.

Note: a Telnet session is not encrypted; so it will transmit your user name, password and other

sensitive or private information in an easily readable format.

Host

This field specifies a hostname or IP address (network node specification) for the remote machine you want to connect to (and which provides the Telnet service).

When you click on the scroll arrow beside the **Host** box, a drop-down box will display host definitions located in your **hosts** file. To select a host, click on an appropriate definition.

Port

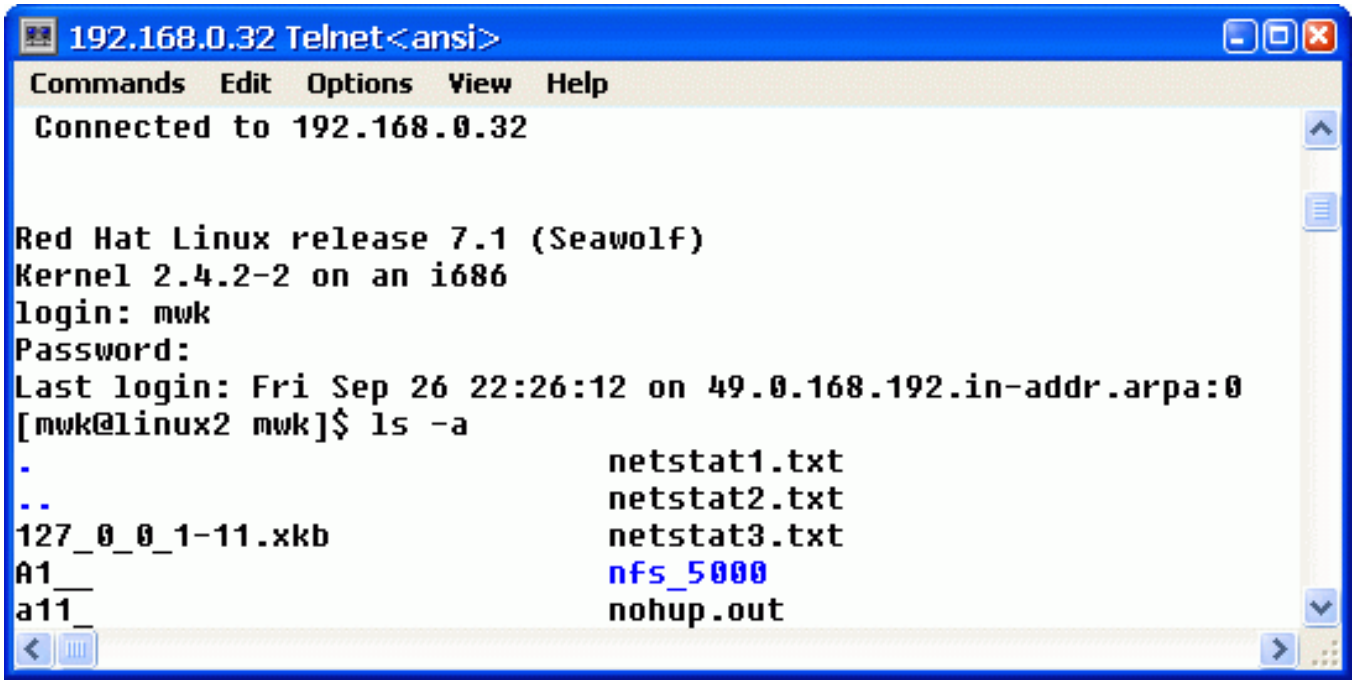
This field specifies the port number of the Telnet service on the remote machine you want to connect to.

The default port number of the Telnet service is decimal 23.

Starting a Telnet Session

To establish Telnet connection, enter the network name or IP address for the **host** you want to connect to, change the default Telnet **port** number if required, and press **OK**. Telnet connects and logs into the specified hostname.

Once you have connected to the host, the host name or IP address you specified appears at the top of the Telnet window (with the terminal emulation mode), and the host login prompt appears in the window:



You must prove your identity to the remote machine using some authentication method (e.g., password authentication). Specify the login information required for your host system. You can then interact with the host by choosing commands from displayed menus, or by typing commands in the window and starting remote applications.

You can customize your Telnet session with the **Settings** and/or **Keyboard Mapping** items in the

Options menu (described below).

The following sequence of commands can be used as an example of working in the Telnet session (depends on the remote shell used):

```
login: arsexam
$ DISPLAY=xtp2:0;  export DISPLAY;
$ xterm&
$ mwm&
```

To capture the screen output of Telnet commands to a file, Telnet writes the log to the **telnet.out** file in the home directory (in case of fatal errors or due to the 'trace' command line parameter).

Terminating a Session

You can terminate a Telnet session by choosing the **Close** command on the Control Menu box, or by selecting **Exit** on the Telnet Commands menu.

If you select **Exit** while a connection to a remote system is still active, Telnet disconnects you from the remote system automatically (properly closing all applications used).

Telnet Menu Options

The Telnet menu bar displays five items: **Commands**, **Edit**, **Options**, **View**, and **Help**. They are described below.

The Help Menu

The **Help** menu contains the following menu commands:

Contents

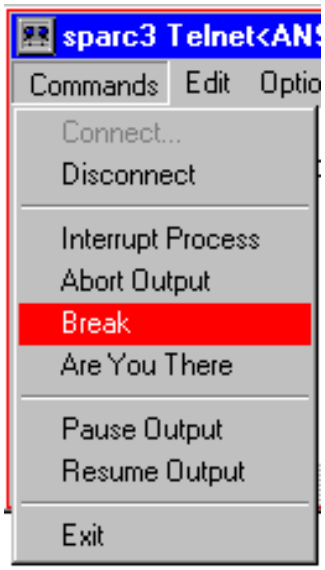
Displays the Telnet help file.

About

Displays copyright, version and program information about Telnet.

The Commands Menu

The **Commands** menu contains the following menu commands:



Connect

The **Connect** item displays the **Telnet Connect Host** dialog box so you can specify the remote system you want to communicate with. You can also connect to a port or service to use other than the standard Telnet port. This is useful when the Telnet client is being used to access something

other than a Telnet daemon.

This command is not available when you are already connected to a remote system.

Once you connect to the remote system, the title bar in the Telnet window shows the remote system name.

Disconnect

The **Disconnect** item ends the connection to a remote system so you can connect to another system or end your session. This command is not available when you are not connected to a remote system.

Interrupt Process

This command sends the Telnet Interrupt Process command (IP control function) to the remote host. This command (which suspends, interrupts, aborts, or terminates the operation of a user process) tells the host to stop the current process to which the terminal is connected. This function is frequently used when a user believes his process is in an unending loop, or when an unwanted process has been inadvertently activated.

Abort Output

This command sends the Telnet Abort Output command (AO control function) to the remote host. This command tells the host to run to completion the current process, which is generating output, but without sending the output to the user's terminal from the host. Further, this function typically clears any output already produced but not yet actually sent to the user's terminal.

Break

This command sends the Telnet Break command (BREAK control function) to the remote host. This command (intended to indicate that the Break Key or the Attention Key was hit) tells the host to stop what it is doing.

Are You There

This command sends the Telnet Are You There command (AYT control function) to the remote host. This command determines if the connection with the host is still up and the system is running. This command tells the host to send back to the user's terminal some visible evidence that the command was received. This function may be invoked by the user when the system is unexpectedly 'silent' for a long time, because of the unanticipated (by the user) length of a computation, an unusually heavy system load, etc.

Pause Output

This command pauses output (it sends **Ctrl+S** to the host). The **Resume Output** item then becomes active and can be selected.

Resume Output

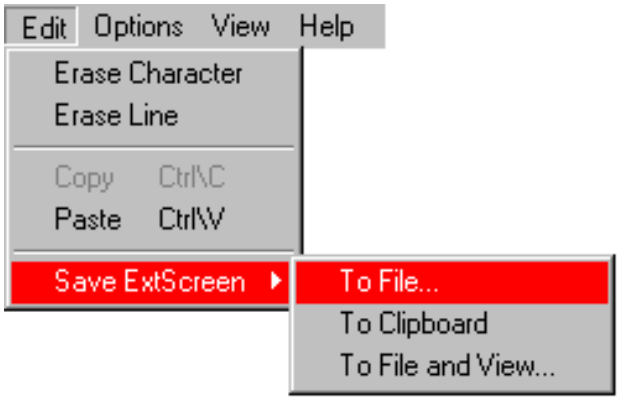
This command resumes output (it sends **Ctrl+Q** to the host) after output has been paused.

Exit

The **Exit** item terminates the Telnet session.

The Edit Menu

The **Edit** menu displays two commands that allow you to edit the lines you type in a Telnet window: **Erase Character** and **Erase Line**. Also there are two standard commands, **Copy** and **Paste**, for text operations with the Microsoft Windows' clipboard.



Erase Character

The host should delete the last preceding undeleted character or **print position** from the data stream being supplied by the user. A **print position** may contain several characters that are the result of overstrikes, or of sequences such as <char1> BS <char2>...

Erase Line

The host should delete all the data in the current line of input, i.e. characters from the data stream back to, but not including, the last **CR LF** sequence sent over the TELNET connection.

Copy

To copy text onto the clipboard, leaving the original text intact and replacing the previous clipboard contents, select the text you want to copy, and choose **Copy**. This command is unavailable until you have selected text.

Paste

When there is text in the clipboard, you can use **Paste** to insert a copy of the clipboard contents at the insertion point to the Telnet window, or to another Microsoft Windows application. This command is not available if the clipboard is empty.

There are several ways in using Copy and Paste. The **first example** will be to show using your mouse.

1. In your Telnet window, scroll over with your mouse to the text you want to select.
2. Then click your left mouse button and hold it down to select the text you want to copy.
3. Once the text has been highlighted in a box frame, right click your mouse button to see the **Copy to clipboard** option, and then click on this button.
4. Your text has now been copied to the Microsoft clipboard, and you can now choose where you

want to paste it.
5. Click on where you want the text to be pasted, and then right click on the mouse and click on the **Paste from clipboard** button. The original text that you copied was now pasted into the area you selected.

The **second example** will be to show using Menus.

1. With the same text to copy, you can highlight the text again. You will see a solid clear box appear around the selected text.
2. Click on the "Edit Menu" and then click on "Copy". This has now copied the selected text to the clipboard.
3. Now click to where you want the copied text pasted to. Click on the "Edit Menu" again and then click "Paste". This will paste the text into your chosen location.

To Copy and Paste, you can also use key combinations, "Ctrl+C"/"Ctrl+V" or "Ctrl_L+Shift_L +C"/"Ctrl_L+Shift_L+V".
For more information, see settings **CtrlCVmode** and **QuickClipBoard**.

See also settings **SelectRect** and **FineSelectMode** that you can use to specify different modes for text selection in your Telnet window.

Save ExtScreen

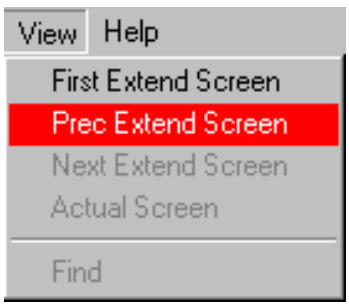
- **To File**
  This option lets you save the output buffer of your session into a file.

- **To Clipboard**
  This option lets you send the output buffer of your session to Clipboard.

- **To File and View**
  This option allows you to save the output buffer of your session into a file and then to invoke Notepad for you to view or edit the text.

  For example, to save a Telnet screen that you want to be written to a file, do the following as you are working in your Telnet window:
  1. Click on "EDIT", then click on "Save ExtScreen" and then click on "To File and View".
  2. Then choose which location you want to save the file to and also what name you want to call the file.
  3. After entering the file name and location where you are saving the Telnet screen, press the Save button. After this the screen shot has been saved. Then you will see a Notepad window with the screen content appear.

The View Menu



With options from this menu, you can navigate through the output buffer of your session. For additional information, refer to subsection **Details of a Session** in section **Starting and Terminating Telnet**.

The **Actual Screen** is the one that always displays the last line of the session's output. In this screen, you can only input your commands.

The **First Extend Screen** option displays lines for the first screen of the output buffer. The **Prec Extend Screen** option displays lines for the preceding screen (if any) from the current one. The **Next Extend Screen** option takes you to the next screen (if any) from the current one.

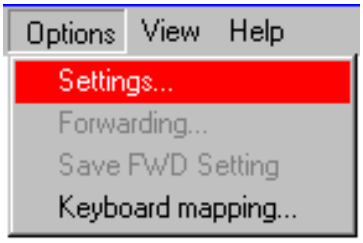
These options are only used for reviewing (not for entering) information and moving to screens. They suspend output into the output buffer and disable your input. To resume output and enable input, use the **Actual Screen** option.

To navigate through the output buffer, you can also use the Notepad-like view-key combinations.

The **Find** option lets you search text in the output buffer like you do that when using Notepad.

The Options Menu

Telnet allows you to personalize your settings and automatically use them every time you establish a connection to a remote computer. This is accomplished by storing your personalized settings in the corresponding ini-files. The Options menu displays items that you can choose to specify particular implementations of Telnet. Normally they do not have to be changed.



- The Settings Option
- The Keyboard Mapping Option

The Settings Option

You can specify the terminal emulation settings for the current connection by choosing the **Settings** item on the Options menu from the main window.

Also, you can click the **Settings** button in the **Telnet Connect Host** dialog box where you can also make some initial settings for your session (that are described in subsection **Details of a Session** in section **Starting and Terminating Telnet**).

The **Telnet Settings** window presents you with a dialog of six tabs that allow you to view and modify the current terminal emulation settings:

- The Keys Tab
- The Type Tab
- The Text Tab
- The Text Tab for the XTERM Type
- The User Defined Tab
- The Logging Tab
- The ExtScreen Tab

The Keyboard Mapping Option

By using the **Keyboard Mapping** Option (i.e. keymap editor invoking), you can load, change (re-define keys and create a new keyboard layout), and save any keyboard definition file.

Keyboard files are text files that define the X Protocol Key Symbols (Keysyms) which are mapped to keys on your keyboard. By default, they have the extension KMF, and are located in the home directory. You specify the KMF file to be used by all XLitePro's programs in the XSettings window.

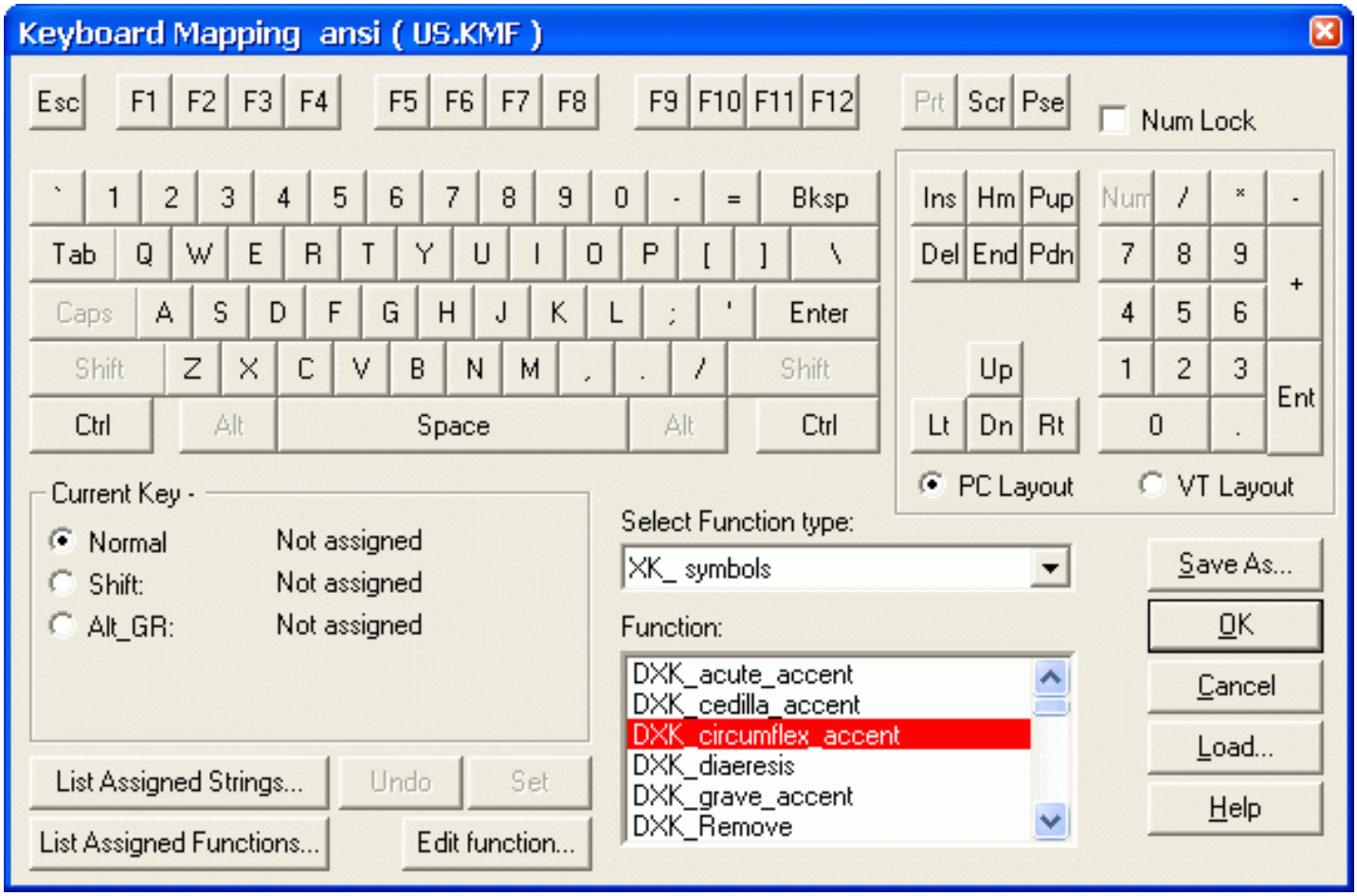
The keyboard mapping file format uses scancodes which allow the terminal to transmit **make and break** codes for each keystroke corresponding to the hardware scan codes used by PC keyboards (scan set 1). **Make** means when the key is pressed; **break** means when the key is released. The Keyboard Mapping File Format is described in Appendix A.

The **Keyboard Mapping** dialog box of the keymap editor allows you to map Keysyms, Characters, or Compose Key Sequences to existing keys on your keyboard.

Keysyms is the encoding of a symbol to a key that exists on a physical keyboard.

Compose Key Sequences are key combinations to produce special Keysyms such as accented characters. These Keysyms are generated by typing two keystrokes. The first key is known as a composing key. Each Compose Sequence consists of two key combinations which generate a new pseudo key.

Modifiers are keys that modify the action of other keys. They are not to be confused with a Keysym. In X Keys they include Shift, Lock, Control, and Mod1 through Mod5. Mod1 through Mod5 are the logical keynames for modifier keys that vary from workstation to workstation. Caution should be used when assigning modifiers to latching keys (NumLock, ScrollLock, or CapsLock). Modifiers mapped to these keys should not be used to modify keys in compose sequences.



The upper portion of the **Keyboard Mapping** dialog box contains a standard keyboard layout. The currently loaded keyboard mapping file name and the terminal emulation mode are displayed at the top of the window.

On the **KeyPad** group box, the KeyPad layout is shown according to the **PC Layout** and **VT Layout** radio buttons states. You can toggle between them to change the KeyPad layout.

NumLock

If you enable the **NumLock** check box, the numeric keypad keys will work as they normally do on your PC (local latched mode). If this option is not checked, the behavior of NumLock is determined by the remote host.

Load

When you press this button, the **Open** standard dialog box to open files appears, allowing you to select and load a keyboard mapping file for viewing and modifying.

Save As

When you press this button, the **Save As** standard dialog box to save files appears, which allows you to save your current keyboard mapping under a new filename.

Select Function type

On this list box, you can select one of the function types: XK_symbol, Character, or Composer to display all values available for it in the **Function** list box.

Function

On this list box, you can select a value for:

- Assigning it to a key with the **Set** button
- Modifying it with the **Edit function** button.

If the Function type selected is **XK_symbol**, a list is displayed containing all of the XKeysyms available.

If the Function type selected is **Character**, a list is displayed containing all of the characters available (including accented characters) with its (decimal/hex) keycode pairs.

If the Function type selected is **Composer**, a list is displayed containing all of the Compose Key Sequences available.

Set

When you click a key on the keyboard layout, it appears in the **Current Key** group box with its current definitions:

- Normal (unshifted/unmodified)
- Shifted
- Modified (with the **Alt GR** key).

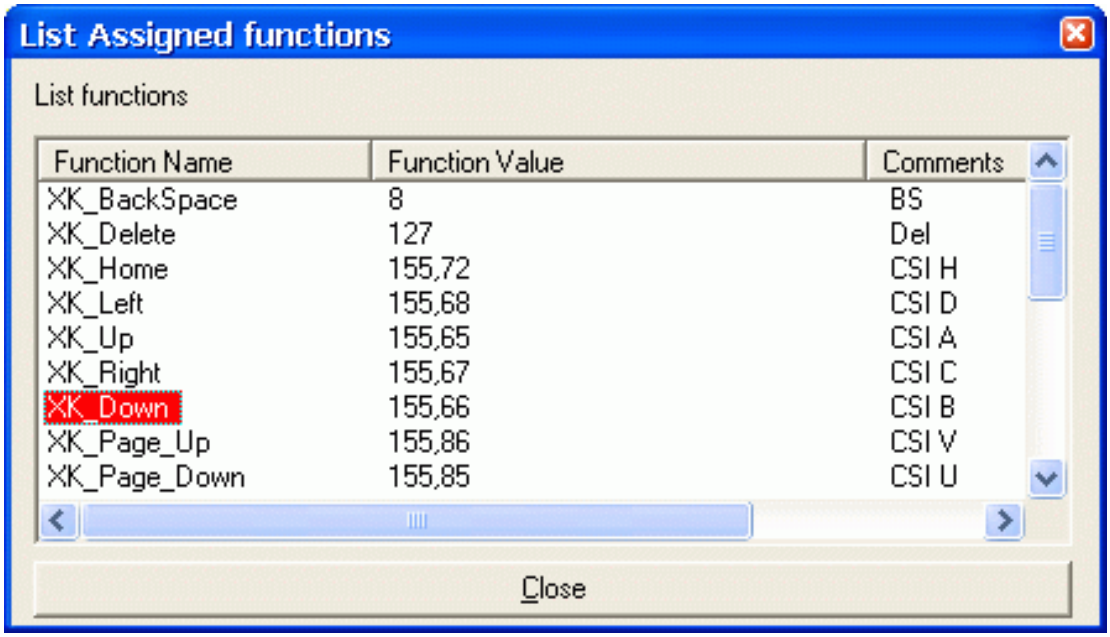
When you have a value highlighted on the **Function** list box (of type: XK_symbol, Character, or Composer) and a key selected on the keyboard layout, you can press the **Set** button to change current values assigned to the key and displayed on the **Current Key** group box to the new value (according to the radio buttons' states).

Undo

Use this button to immediately restore the previous key value every time you press the **Set** button.

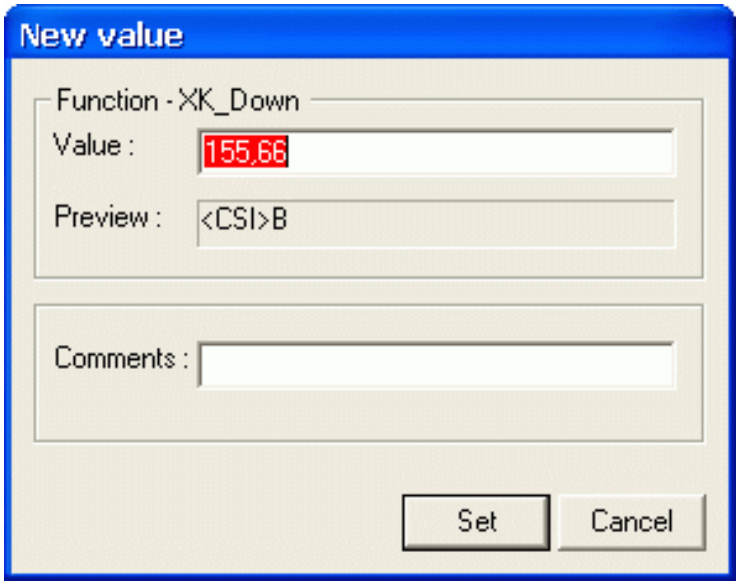
List Assigned Functions

When you press this button, a dialog box appears that allows you to view a list of functions already assigned to functional keys (for the current terminal emulation mode). The list contains function names (X Keysyms), function values (code sequences), and comments on them.



Edit function

When you have a highlighted function of either the XK_symbol or Composer type, you can press the **Edit function** button to change the value to define a new key sequence for the function (and current terminal emulation mode). The **New value** dialog box will appear on your screen.



The **Function** group box shows the currently selected function.

Value

This edit field is used to enter a new string for the selected function. The string can include decimal codes (in the range of 0...255) separated with the comma character (as in the **List Assigned Functions** dialog box). The string should be in valid KMF format described in Appendix A.

Preview

This field displays a comment value for a selected function.

Comments

Use this field to enter a new comment for the function you define.

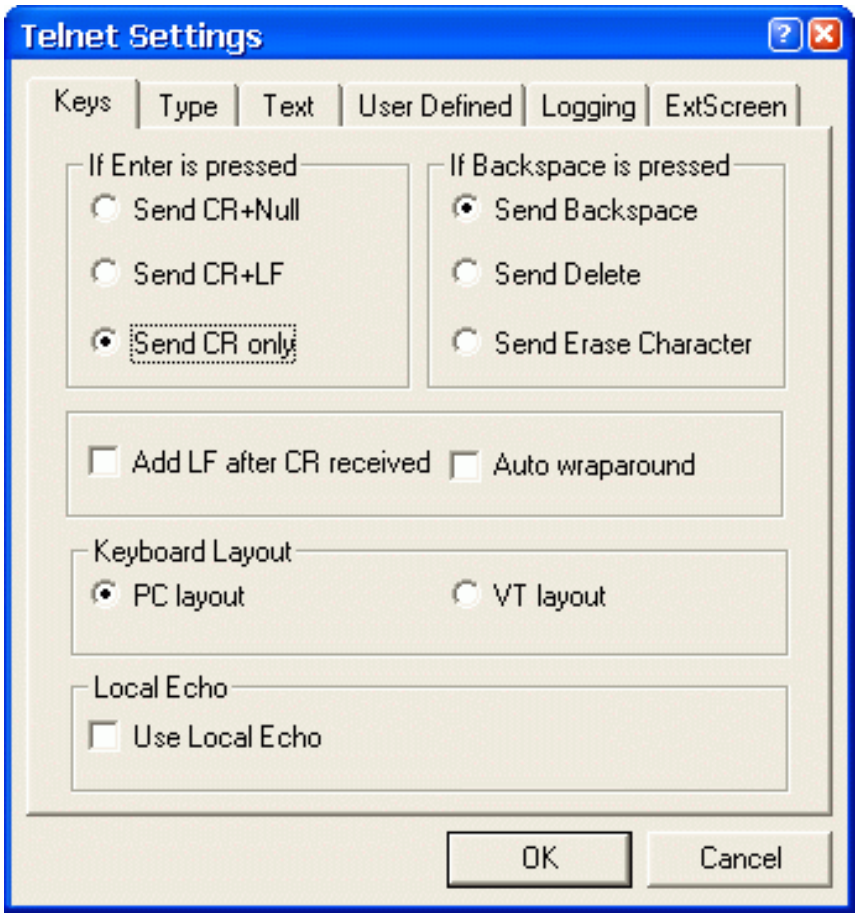
Set

This button stores new values you entered and exits the dialog.

Cancel

You can cancel any changes you made to the dialog box by clicking on this button.

The Keys Tab



If Enter is pressed

Options in this group box define the end-of-line sequence sent when you press the **Return** or **Enter** key.

If Backspace is pressed

Options in this group box specify whether the **Backspace** key will be interpreted as **Erase Character**, **Backspace**, or **Delete**.

Add LF after CR received

This option allows you to modify (or not) the **CR** code received over the network.

Auto wraparound

If this check box is enabled, input text will be automatically wrapped on the next line when your string is too long (i.e. any characters received when the cursor is at the right margin will be

displayed on the next line).

Otherwise, input is stopped so you cannot enter more characters (i.e. any characters received when the cursor is at the right margin will be displayed just to the left of the right margin, replacing the current character displayed there).

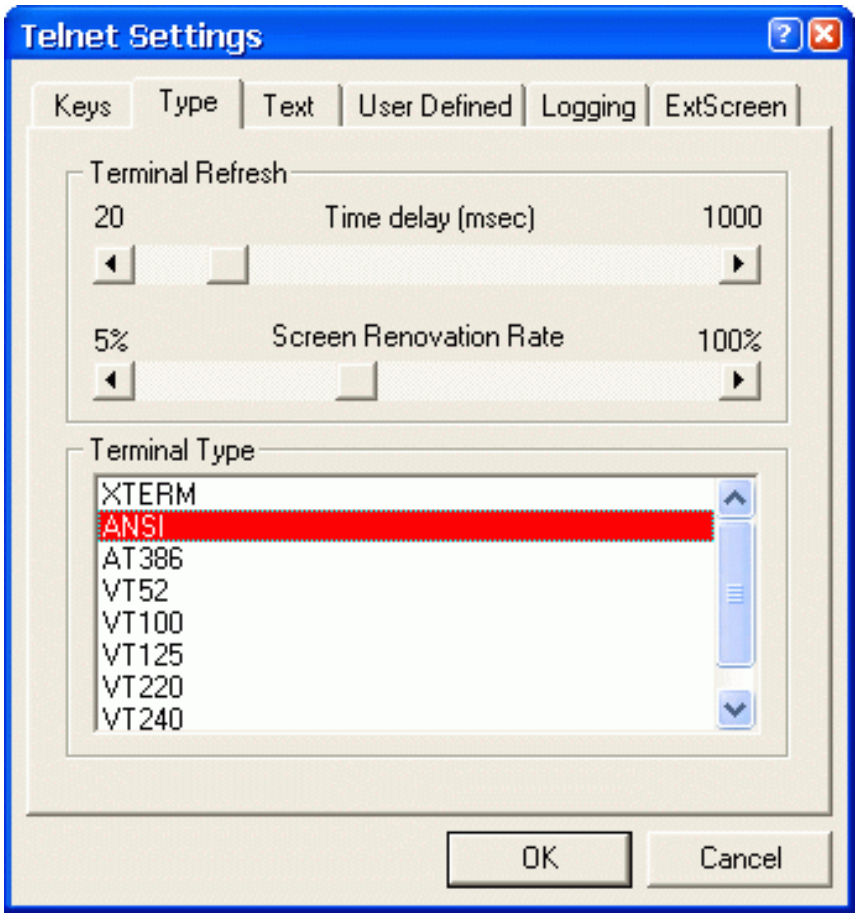
Keyboard Layout

Options in this group box specify which keyboard layout will be used: PC layout or VT layout.

Local Echo

Some Telnet daemons may not support the standard Telnet protocol (and do not send a symbol of a pressed key). With the **Use Local Echo** check box enabled, you can display a key character regardless response from that Telnet daemon.

The Type Tab



Terminal Refresh

This group box allows you to change values of parameters that control the screen buffer output and modify the characteristics of your keyboard.

The **Time delay (msec)** parameter sets the time interval (20...1000) that defines when to display lines with character(s) received.

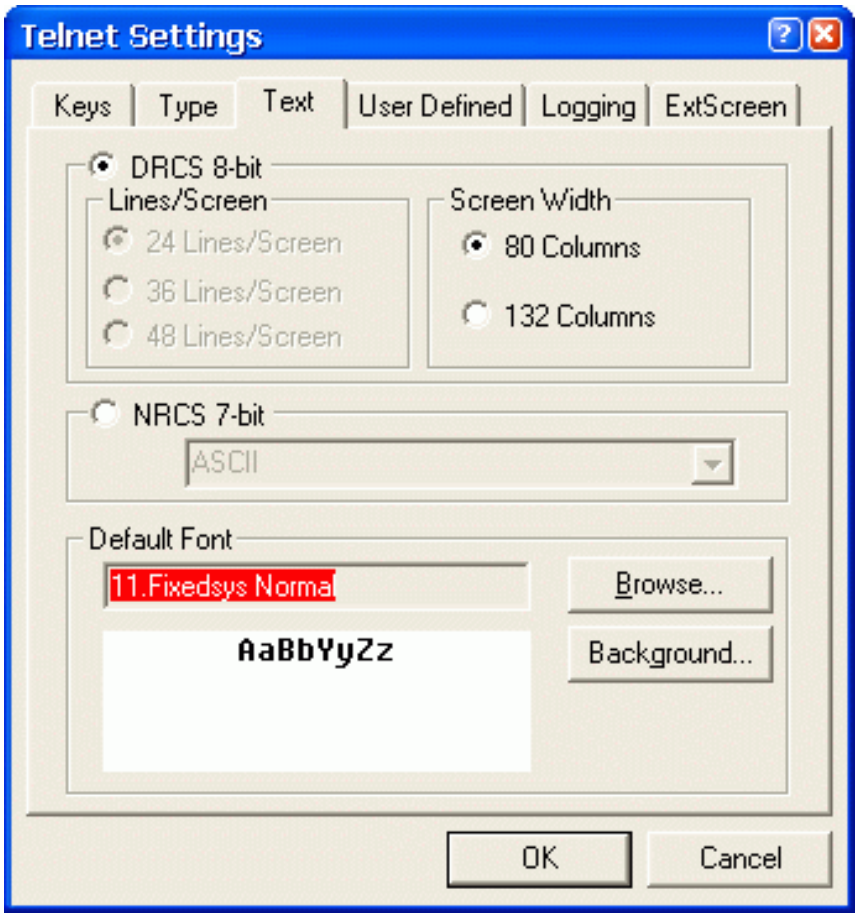
The **Screen Renovation Rate** parameter sets the ratio (5%...100%) of screen changes (e.g. characters entered or modified) to full screen that defines when the screen area modified will be re-displayed.

Terminal Type

This option allows you to change emulation modes for the Telnet session by selecting one of the available modes from the **Terminal Type** list. The mode must correspond to that assigned in the TERM() command when logging in. Telnet adjusts your system so that your computer, keyboard, and terminal perform just as the specified terminal does. The modes are popular control sets used in terminals originally manufactured by Digital Equipment Corporation (DEC). If you are not sure which

terminal to select, select VT-100 (ANSI escape sequences).

The Text Tab



You can specify the lines of text that you want to be retained in memory so that you can scroll through it in the window. Options in this group box let you specify the number of lines (24/36/48) and columns (80/132) that will appear in the Telnet window.

DRCS 8-bit

When this radio button is enabled, you define to work in the 8-bit environment and send the 8-bit control sequences and graphic characters (Multinational character transmission mode), including supplemental characters.

In this mode, you can download soft character sets from the host system into the terminal. The soft character set is also known as a **dynamically redefinable character set** (DRCS). This feature lets you design your own soft character sets for use with the terminal.

You can use the DECDLD control string command to down-line load one or more characters of a specified 94- or 96-character DRCS with a specified logical pixel pattern.

NRCS 7-bit

When this radio button is enabled, you choose to work in the 7-bit environment only. Select one of the 7-bit character sets from the **National Replacement Character Sets** (NRCS) pull-down list box to allow for country/region's replacement characters to be sent in the 7-bit escape/control sequences (National character transmission mode).

The following NRC sets are available:

- ASCII
- DEC Special Graphics
- DEC Supplemental
- British
- Dutch
- Finnish
- Norwegian/Danish
- Swedish
- French
- French Canadian
- German
- Italian
- Spanish
- Swiss

In VT100 mode, VT52 mode, or when **7-bit NRCS characters** is selected (through Set-Up or the DECNRCM command), only ASCII, NRC sets, and DEC Special Graphic characters are available.

Default Font

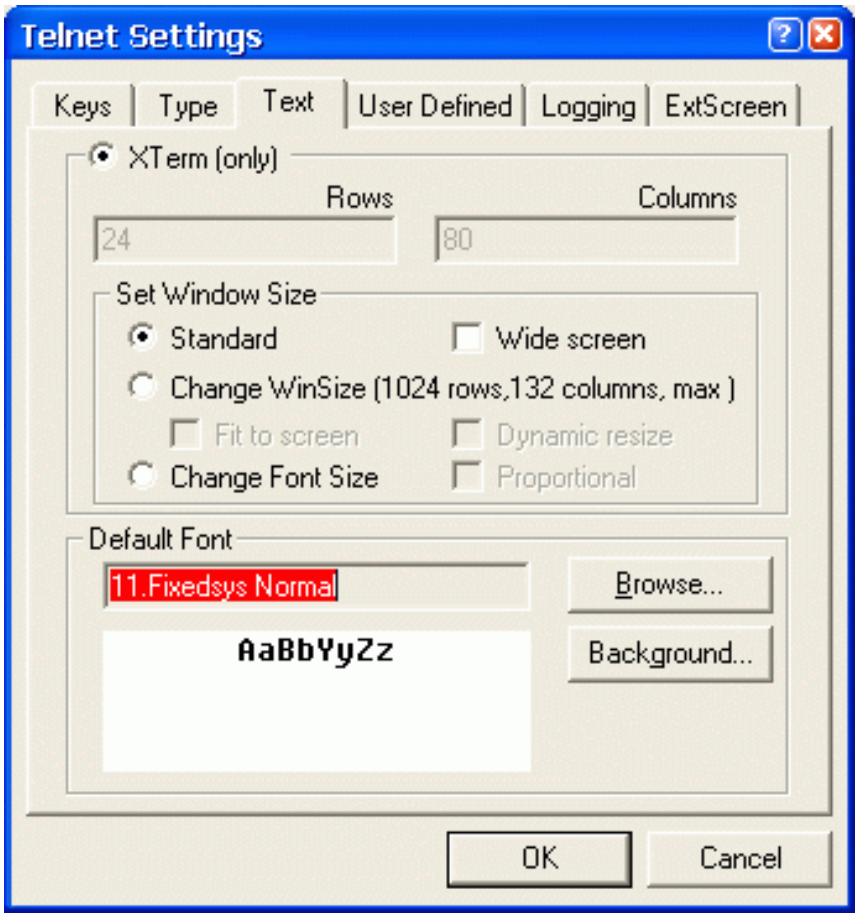
Characters in the Telnet window appear in the specified font, size, and colors. Options in this group box allow you to change font parameters used to display text in the Telnet window.

Browse

When you press this button, the Font standard dialog box appears. This dialog box changes the font name, style, and size of text displayed in the Telnet window. Also, you can choose a desired font script, color, and effects (strikeout and underline).

When you press the **Background** button, the Color standard dialog box appears. You can define your color for your background. The Colors tab allows you to customize the color of your screen by emulating the color of the host's attributes. The colors you set in this tab are not altered by the colors settings you make in the Windows Control Panel.

The Text Tab for the XTERM Type



You use this tab to select font and specify screen size. Also, you can choose one of the three modes to define appearance of your font against window size (for the XTERM terminal type only).

The XTERM terminal emulation provides the 16-colors mode instead of B/W one.

Rows

This field lets you specify the number of rows that will appear in your emulation window.

Columns

This field lets you specify the number of columns that will appear in your emulation window.

The Set Window Size Box

In this box, you can select one of the three modes to define appearance of your font against window size.

- **Standard**
  The row and column numbers define the size of your actual screen (see section **The View Menu**). The values are read in from the ini-file (first from the terminal definition file, **Terminfo.ini**).

  In this mode, your initial window size will only be defined by the screen size and font size selected. The window displays scrollbars when its size is less than that of the screen.

  The **Wide screen** check box toggles the number of columns between 80 and 132.

- **Change WinSize**
  The row and column numbers you specified define the size of actual screen (see section **The View Menu**). You may type in up to 1024 rows of 132 columns each.
  In this mode, font appearance will not be changed when you change your window size. The window displays scrollbars when its size is less than that of the screen.

  With the **Fit to screen** check box selected, an application window is fully displayed on the PC's screen (i.e., it does not cross screen boundaries). If necessary, Scroll Bars are created.

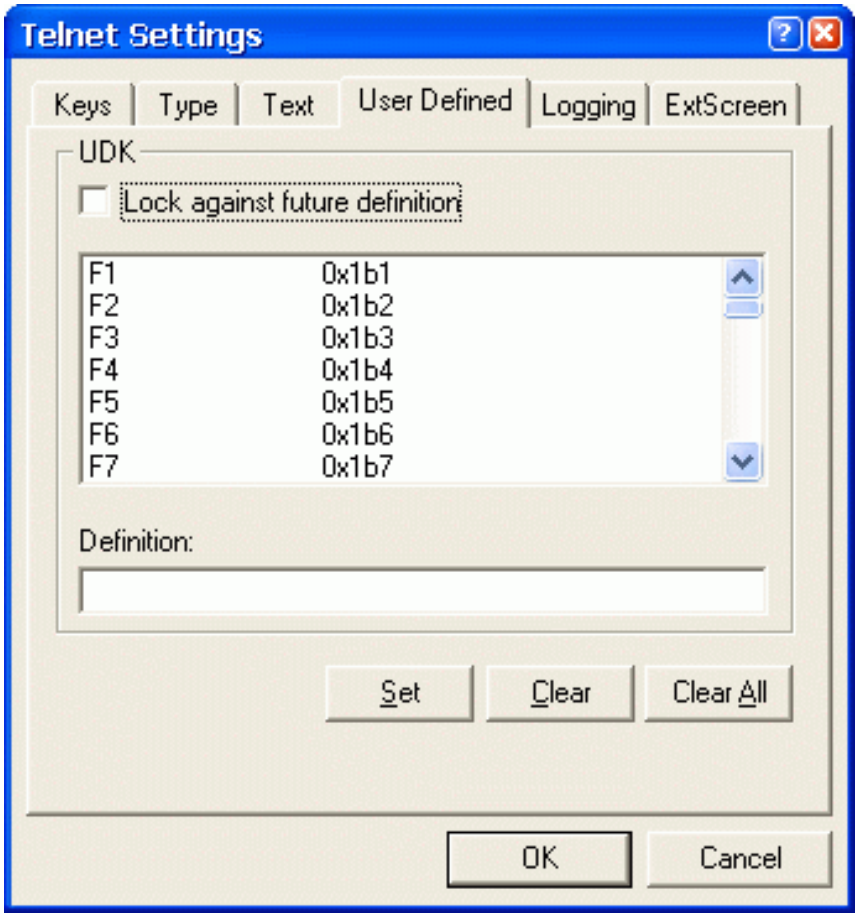
  With the **Dynamic resize** check box selected, the "no scrollbar" mode is provided. The actual Rows and Columns numbers are dynamically being changed while window resizing.

- **Change Font Size**
  The row and column numbers you specified define the size of actual screen (see section **The View Menu**). You may type in up to 1024 rows of 132 columns each.
  In this mode, font appearance will be changed according to your window size. The window displays the entire screen.

  With the **Proportional** check box selected, the vertical dimension will only define font appearance. Otherwise, both dimensions of your window will define it.

The User Defined Tab

This tab allows any functional key to be programmed with a user-defined sequence. User-defined keys (UDKs) are a subset of functional keys.



The UDK group box contains a list box with currently defined keys for a current emulation mode. This box allows you to map key symbols to the Unshifted, Shifted, Mode Switched, and Shift-Mode Switched states of the key. You can select a key symbol and then clear (with the **Clear** button), define or re-define its function value (in the **Definition** edit field).

You can use UDKs like a macro defined for a functional key: whenever you want to forward a user-defined control string to a host you press the key combination to activate the value. (Also see the **List Assigned Functions** dialog box in the **Keyboard Mapping** option below for already defined functional keys.)

Note: some function key combinations are reserved by MS Windows and cannot be redefined.

Upon terminating Telnet sessions or pressing OK, UDKs are stored in the **terminfo.ini** file (in the emulation mode section; see Appendix B for details), so they will be defaults for the next session when the file will be read in.

Lock against future definition

Use this check box to lock/unlock UDKs listed against future redefinition (from a remote host).

Definition

This edit field is used to enter new control string codes for UDKs. The string can include any combination of escape sequences, control sequences, or text (without any separating character). The string should be in valid format for the terminal emulation mode. You can scroll the field left or right as needed to allow longer strings to be entered.

Set

This button assigns the value entered in the **Definition** field to the UDK currently selected in the list box (for the current terminal emulation mode). This key combination will activate the value whenever it is pressed.

Clear

This button removes a value for a currently selected UDK.

Clear All

Click this button to delete the mapping for all UDKs listed.

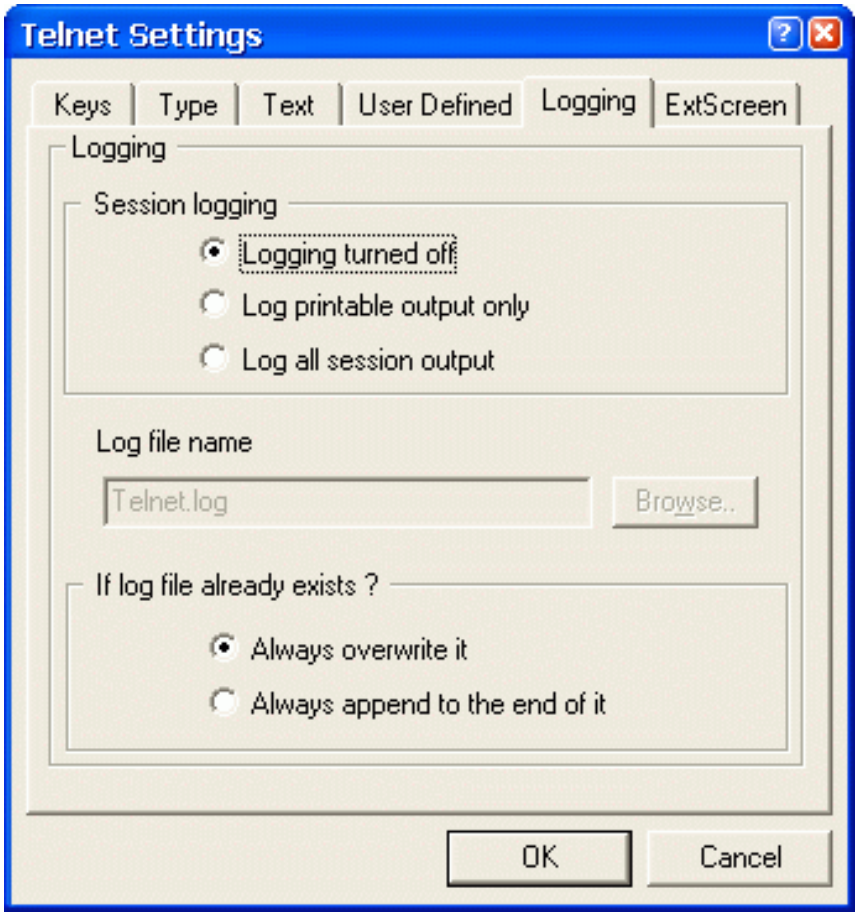
OK

Pressing OK saves current UDK settings and quits the dialog box.

Cancel

You can cancel any changes you made to the dialog box by clicking on this button.

The Logging Tab



The Session logging Box

These radio buttons control the amount of log output you store in a log file for your session.

Logging turned off

With this option on, no log file is used and no log output is stored.

Log printable output only

With this option on, only printable log output is stored in a log file you specified (i.e., without escape sequences).

Log all session output

With this option on, all log output is stored in a log file you specified (i.e., with escape sequences).

The Log file name Box

This is the actual setting if logging turned on.

In the entry field, you can enter a name for the log file to store data for the session.

The **Browse** button can help you locate and select a desired file.

The If log file already exists Box

These are the actual radio buttons if logging turned on.

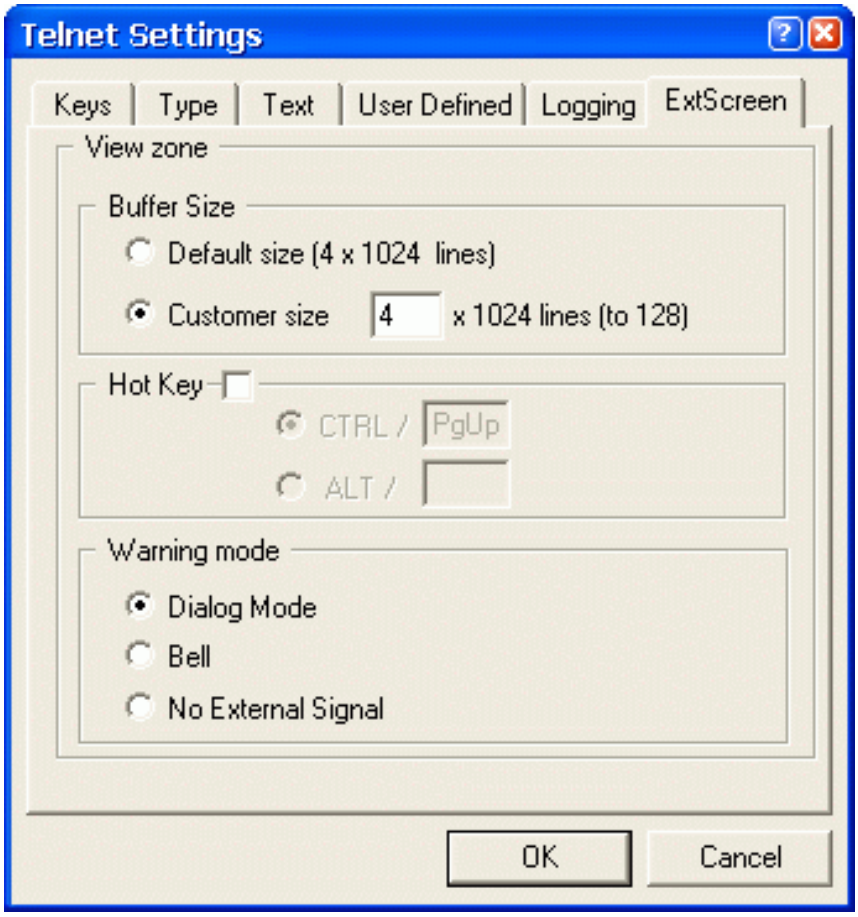
Always overwrite it

With this option on, a new log file for your session will overwrite one that already exists.

Always append to the end of it

With this option on, a new log data will append to the end of the log file that already exists.

The ExtScreen Tab



The Buffer Size Box

In this box, you specify the amount of memory that will be allocated and used for output.

Default size (4 x 1024 lines)

With this option on, the default size for the output buffer will be used.

Customer size

With this option on, you must specify a number of blocks (each of 1024 lines) for the size of the output buffer that will be used for your session.

Hot Key

With this check box selected, you can choose a hot key combination you will use to change to

navigation mode to view the output buffer.

Warning mode

These radio buttons control the way the program will let you know that you change to the actual screen while navigating through output buffer.

Dialog Mode

With this option on, a dialog box will appear.

Bell

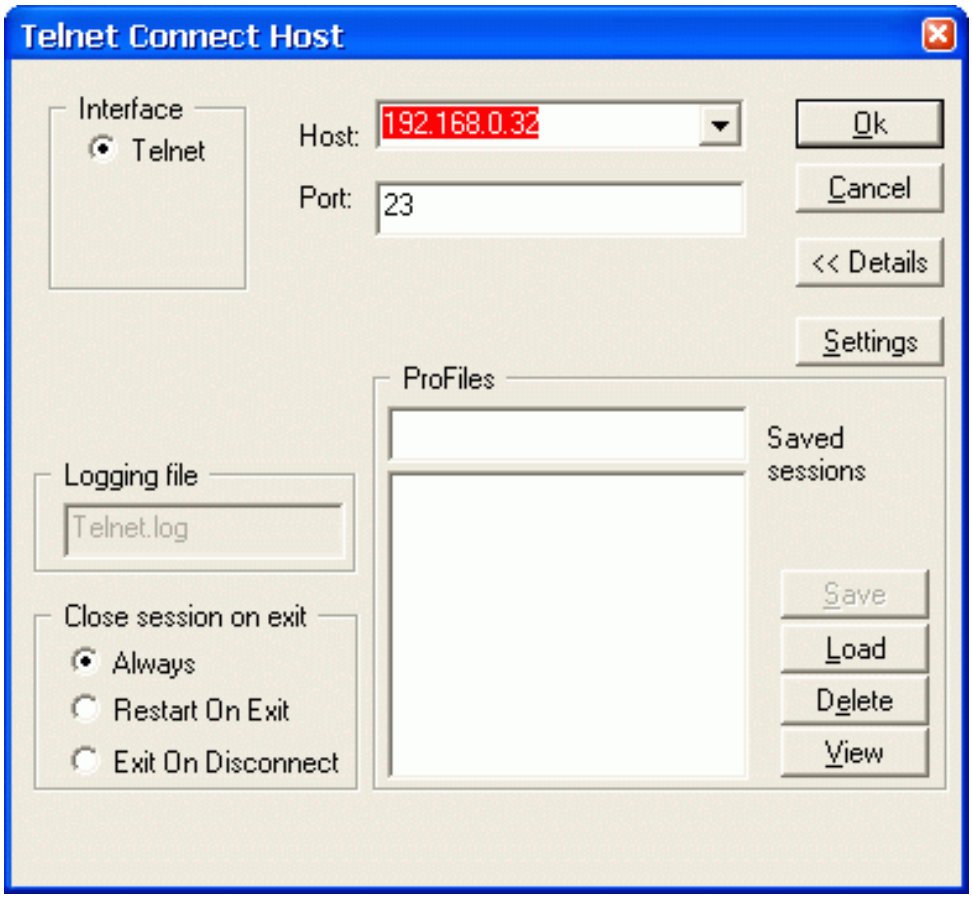
With this option on, a system bell will sound.

No External Signal

With this option on, the program will not warn you that you change to the actual screen while navigating through output buffer.

Details of a Session

Clicking the **Details** button in the **Telnet Connect Host** dialog box will change it into the following form:



Clicking the **Details** button once more will change the dialog box into the short form.

Clicking the **Settings** button will display the **Telnet Settings** dialog box. (Refer to subsection **The Settings Option** in section **The Options Menu**.)

Pressing **OK** will store current settings (for the next session) and will establish a connection, using them.

You can cancel any changes you have made to the dialog box by clicking on **Cancel**. This will also close the dialog.

The Logging file Box

In the entry field, you can enter a name for the log file to store data for the session.

The Close Session on Exit Box

In this box, you choose a mode of closing your session.

Always

In this mode, the session will always be terminated on exit or on Disconnect.

Restart on Exit

In this mode, the session will be restart on exit (i.e., by displaying the **Telnet Connect Host** dialog box).

Exit on Disconnect

In this mode, the session will be closed on Disconnect.

The ProFiles Box

In this box, you can manipulate with profiles (i.e., saved session settings).

In the ProFiles list, names of available profiles are displayed. In the entry field above, the name of the last loaded (by **Load**) profile is displayed.

When you start up the program, the settings of the last run are loaded from the ".." profile and are made as current settings. On closing a session, its current settings are automatically stored in the ".." profile. You can **Save**, **Load**, and **View** this profile, but cannot **Delete** it. Note that **Load** will change its contents.

Save

In the entry field, you can type in a name for the profile you want to save. The **Save** button saves current settings in the profile specified and adds its name to the Profiles list.

Load

This button loads the profile you select from the Profiles list, makes its settings as current ones, and displays its name on the entry field.

Delete

This button will delete the profile selected from the Profiles list.

View

This button displays in Notepad the profile (i.e., its set of settings) you select from the Profiles list.

The "[TELNET]" Section of the ini-file

The "[TELNET]" section of the ini-file may have the following entry lines you can customize for particular needs and applications.

ALTforEMACS=1

This setting allows the ALT key to be used in EMACS.

ExitOnDisconnect=N

This option is used to control the session's restart/completion mode, with **N** providing the following:

- **ExitOnDisconnect=1**
  suppresses starting the new Telnet session;
- **ExitOnDisconnect=2**
  suppresses issuing the "Connection lost" message (only writing it into the **telnet.out** file);
- **ExitOnDisconnect=3**
  suppresses issuing the "...open forwarded connections..." message (only writing it into the **telnet.out** file).

FineSelectMode=1

This setting is used to set up the **Fine Selection** mode. In this mode, all "touched" symbols are included in a selection area.

QuickClipBoard=1

This setting is used to set up the **Quick Copy to Clipboard** mode. In this mode, a selected area is copied to Clipboard immediately on release of the left mouse button.

SelectRect=0

This setting is used to set up the **Selection-by-filling** mode that provides text highlighting selection in traditional manner.

CtrlCVmode=N

With **N=3**, this option specifies to use the traditional key combinations, "Ctrl+C"/"Ctrl+V", for **Copy/Paste** operations in the session.
Caution: in this mode, you cannot send another "traditional" key combination, "Ctrl+C" (i.e., Break Event), to remote applications.

With **N=5**, this option specifies to use the default key combinations, "Ctrl_L+Shift_L+C"/"Ctrl_L+Shift_L+V", for **Copy/Paste** operations in the session.

Running Telnet with Command Line Parameters

The 'host' parameter

You can launch Telnet with the 'host' command line parameter to avoid interactive input of it. 'Host' may be either a host name or IP address for the remote machine you want to connect to.

Examples:

PATH\Telnet.exe u2-1

PATH\Telnet.exe 192.168.136.223

where PATH indicates your XLitePro home directory.

The 'xini' parameter

You can launch Telnet with the 'xini' command line parameter:

PATH\Telnet.exe -xini <IniFilePath>

where <IniFilePath> specifies a full path to a specific ini-file.

This feature allows you to run several Telnet sessions each with its own **xwp.ini** file (i.e. settings).

In order to do so, you can create a new Telnet shortcut (e.g., in the XLitePro Programs' folder) and fill in the **Target** field in Properties of it with the command. For example, the **Target** field in Properties of the Telnet shortcut might contain the following line:

"PATH\Telnet.exe" -xini myxwp1.ini

By default, the field contains a call of Telnet with no arguments, and the **xwp.ini** file will be used in this case.

To create your specific ini-file, you can copy the **xwp.ini** file and then change required parameters with the Telnet utility by starting it with the command line parameter.

The 'trace' parameter

You can launch Telnet with the 'trace' command line parameter

PATH\Telnet.exe -trace

to collect debug information in the **telnet.out** file in the home directory.

The 'lxdn' parameter

The "-lxdn [<DisplayNumber>]" command line parameter is used to set up the DISPLAY session environment variable if a remote daemon supports it. This provides correct X11-forwarding (if set).

The "-lxdn" option means looking up running XServer (i.e., Telnet understands the local XServer's "DynamicDisplayNumber" mode and correctly looks up running XServer).

The 'mout' parameter

The "-mout [<outputfile>]" command line option is used to provide a "readable" text-tracing to the "outputfile" log-file for your session.

The 'log' parameter

The "-log <LogLevel>" option is used to set up the level of tracing (0..7).

The 'prof' parameter

With the "-prof profilename" command line option, you can initialize a session that will first read in the profile specified and then use its data for the session.

Terminal Emulation in Telnet

The Telnet program can emulate XTERM, AT386, ANSI, VT52, VT100, VT125, VT220 and VT240 terminals. The **terminfo.ini** file describes the capabilities of these terminals. This description contains control sequences for them and is very similar to the TERMINFO source code of the UNIX system. So users can edit the file to suit to the special environment.

By editing the **terminfo.ini** file, users can define the terminal type, the screen size (the number of lines and columns), the number of colors, the color palette (i.e. RGB values for each color number), sequences to be transferred to remote hosts for each user-defined key on the keyboard.

Appendix B contains detailed information on how to describe the terminal emulation capabilities.

8. Startup

This chapter describes how to use the Startup application supplied with XLitePro.

Startup is a program for automating host access with using the REXEC, RSH, RLOGIN, or Telnet protocol. REXEC and Telnet operations may be done either by direct access to a remote host or through one of the established SSH1/SSH2 protocol connections as well.

By using the Startup program, you can:

- Enter one or more commands in a single line and execute them on a host
- Run a local startup file (with a sequence of executable commands) on a host
- Enter a command line to run a local executable file (e.g., telnet, local X clients, etc) and execute it on your PC
- Create/save/select/remove/open/execute startup jobs.

Startup job is a task with a certain set of parameters (start method, login information, command line, settings). You can create a job (i.e. store current parameters under a certain name). A job may be launched by clicking on its icon.

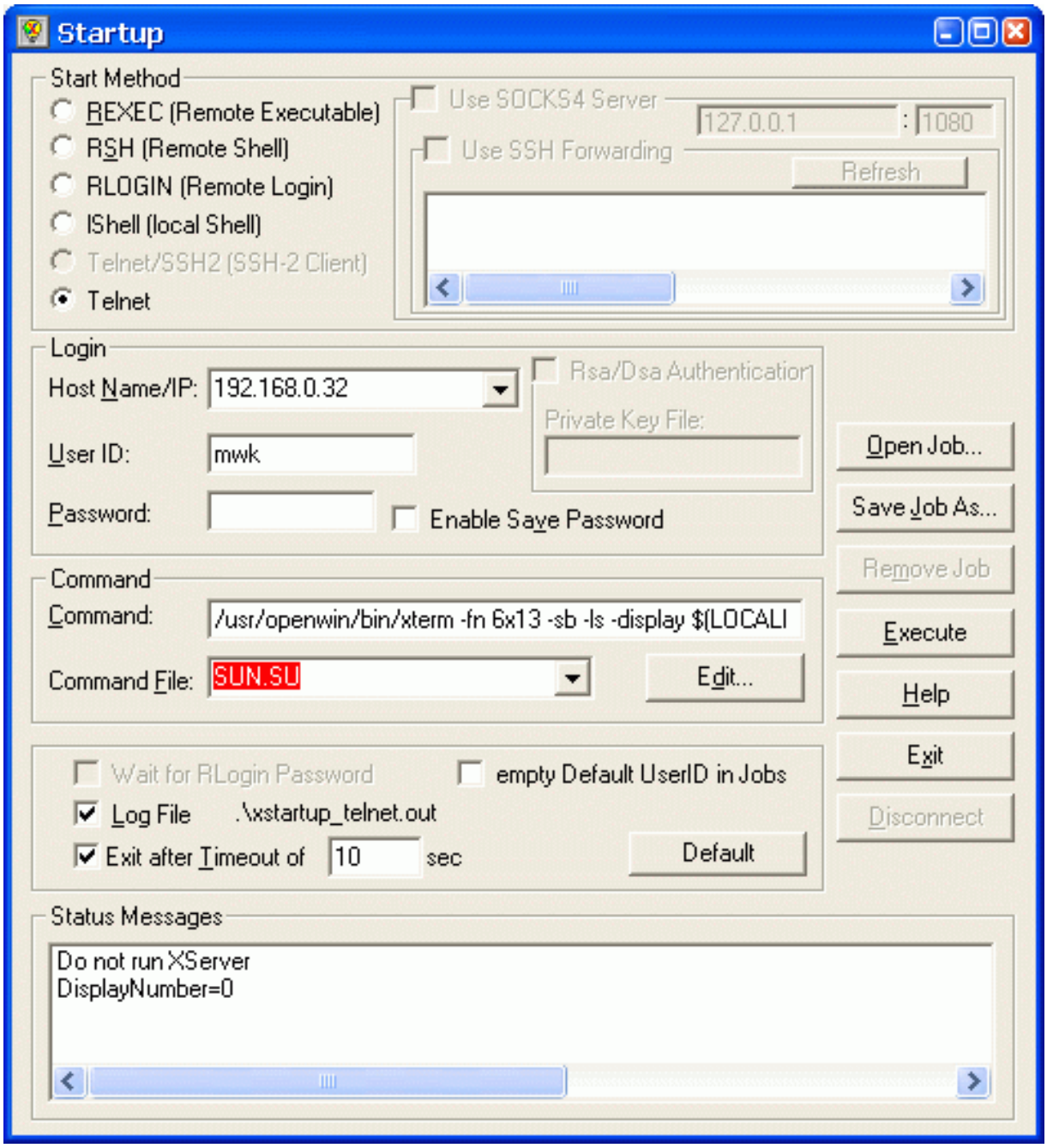
In order to use Startup, your host system must support the REXEC, RSH, or RLOGIN protocol. Besides, you need to know the following about any host you want to connect to:

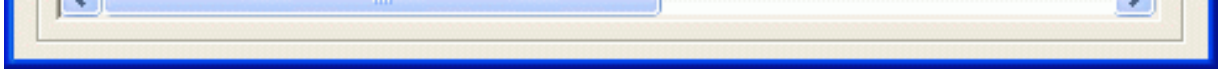
- For REXEC or RLOGIN method, your user ID name and password.
- For RSH method, your user ID name. Some hosts require your PC to be authorized before using RSH, as it does not require a password. On UNIX systems, you need to add your PC name to the **/etc/hosts.equiv** and **$HOME/.rhosts** files on your host.
- Either the network name or address of the host.

Starting and Terminating Startup

You can start the Startup program by double-clicking on the **Startup** icon in the XLitePro Programs' folder:

The **Startup** window will appear on your display:





You can then enter information required to run remote commands into the window (see **Entering Startup Info** below).

Once you have done this, you can execute commands or run startup files on a host as it described in the next section.

In order to terminate the Startup program, click on the **Exit** button.

Running a Startup File

A startup file consists of one or more ASCII lines. Each line may contain one or more executable commands on the host. Use your text editor to create the file in the XLitePro's home directory. All startup files have to have the **.SU** extension.

In order to execute instructions from any startup file, specify a name for the file you want to run and press the **Execute** button (see the **Command Box** section below).

Executing a Command/File

When you execute a command or startup file (remotely), the following takes place:

- The login information in the **User ID**, **Host**, and **Password** fields will be used to make a connection.
- The command in the **Command** field or in the startup file will be sent to the host with the name or IP address specified in the **Host** field. If the command starts a client, the client session will commence. The initial socket will be closed after the amount of time specified in the **Exit after Timeout of** field. Note that different lines of a startup file will spawn different processes.
- If **Log File** has been checked, host or command messages will be saved in a log file for the amount of time specified in the **Exit after Timeout of** field.

Entering Startup Info

This section describes the Startup fields you can use to select or enter proper parameters to run remote commands.

The Start Method Box

The first thing you should select is the **Start Method**. Initially **REXEC** is selected.

RSH

For RSH method ("Remote Shell"), you should know your user ID name. If you click on **RSH**, the **Password** field will be grayed. Some hosts require your PC to be authorized before using RSH, as it does not require a password. On UNIX systems, you need to add your PC name to the **/etc/hosts.equiv** and **$HOME/.rhosts** files on your host.

REXEC

For **REXEC** (Remote Executable) method, you can choose either the direct access insecure execution mode or the secure (SSH) channel mode.

In the Start Method box of the Startup window, you can see a list of actual SSH1/SSH2 protocol connections you have previously established (for example, by the SSH1 or SSH2 Client of the Telnet_SSH program). The **Refresh** button allows you to renovate the list.

If you clear the **Use SSH Forwarding** check box then the direct access insecure mode will only be used for REXEC operations.

If you want REXEC operations to be done in secure connection mode, enable the **Use SSH Forwarding** check box and select a desired remote host (an active SSH server) from within the list.

If the REXEC server and SSH1/SSH2 server you are using in a Startup session are on the same remote host then the **Host Name/IP Address** field can contain "localhost" or "127.0.0.1". This is more preferable than its actual IP address in most cases (but depends on how the host has been configured).

You can use the netstat utility to watch your network connections.

RLOGIN

For **RLOGIN** (Remote Login) method, you can use the following macros in a command or startup files (see section **Command Box** below):

- "$(DELAY)"
- "$(CR)"
- "$(LF)"
- "$(CRLF)"
- "$(SEND)"
- "$(INPUTPASSWORD)"
- "$(DISPLAYNUMBER)"

Note that the **profile** file is always executed on a remote host for this method when a user logs in.

Startup uses the randomized RLOGIN initial portnumber to avoid possible rejection of connection.

IShell (local Shell)

The **IShell** (i.e., local Shell) mode allows you to enter a command line to run a local executable file (on your PC) like you do that with the MS Start/Run box. For example, you may choose to run Telnet, Notepad, local X clients, etc.

For information about local X clients, please refer to Appendix F **Local X and Motif Clients.**

Telnet

The **Telnet** method is similar to other methods. You enter login information to establish Telnet connection with your remote host and specify a command line or file you want to execute on the host. When you press **Execute**, Startup will launch a non-interactive Telnet session (i.e., without the **Telnet** window).

In this mode, the direct access insecure mode will be used for Telnet operations (without SSH X11 forwarding).

Note that you must use the **-display** argument in the command line to invoke a remote X client (to communicate with XServer on your PC-host). In this case, the X client connects to your XServer and operates.

For example, you can execute the following command line to invoke the 'xterm' application on your PC-host:

path/xterm **-display** 192.168.36.41:0 &
or
path/xterm **-display** $(LOCALIP):$(DISPLAYNUMBER) &

The Login Box

The **Login** box lets you enter information required to log onto a host. If your start method is set to **RSH**, the **Password** field is disabled.

Host Name/IP Address

This field defines a name for a remote host you want to connect to and execute commands. You can enter either the network node name for the host or its network address. When you click on the scroll arrow beside the **HostName** field, a drop-down box will display host definitions being in your **hosts** file. To select a host, click on an appropriate definition.

Note that a host name you specify here may not result in the same remote host name due to different name-address DNS translations in different LANs. You should specify the host name or IP address for a remote host in a LAN that are known inside that LAN for that host.

User ID

This field is where you enter your **User ID** for the host you want to connect to.

Password

This field defines your password for the REXEC, RLOGIN, or Telnet method. When you enter your password, an asterisk will appear for each character you type in.

Enable Save Password

If this check box is disabled, then you will be prompted to enter a password each time when it is needed (e.g., when you **Save Job As** with the setting and then launch the job).

The Command Box

The **Command** box lets you enter information required to execute a command or to run a startup file on the specified host.

Command

This field is used to enter a command that will be sent to a host or to specify a startup file you want to run. The command you enter depends on your host system and how it is set up. In the **IShell** mode, you should enter a command to be run locally (on your PC). You can enter up to 255 characters in a command line.

UNIX C Shell Command

If you have defined all necessary environment variables in the C shell resource file (i.e. the **.cshrc** file), then you can simply enter a command. For example, to start **xterm**:

xterm &

UNIX Bourne Shell Command

Because the Bourne shell startup file (i.e. the **.profile** file) will not be invoked when the Startup program logs onto the host, it is recommended that you include the **path** and **display** variables on the command line. For example:

path/xterm **-display** mypcname:0 &

where **path** is the location of the **xterm** on the host, and **mypcname** is the network name or IP address of your PC.

Note that you can use the following macros in a command or startup files:

- $(LOCALHOST) for the network name of your PC
- $(LOCALIP) for the IP address of your PC
- $(LOCALDISPLAY) for the display specification of your XServer
- $(DISPLAYNUMBER) for the display number of your XServer

For example:

path/xterm **-display** $(LOCALHOST):0 &

path/xterm **-display** $(LOCALIP):0 &

path/xterm **-display** $(LOCALDISPLAY) &

path/xterm **-display** $(LOCALIP):$(DISPLAYNUMBER) &

Also, you can use macros $(INPUTLIP) to place any text into the command line before its executing. For example, if the **Command** field contains the following line:

xterm -**display** $(INPUTLIP):0 &

then you will be prompted to input the text string (an IP address is assumed). If you enter 1.2.3.4, then the resulting command line will be:

xterm -**display** 1.2.3.4:0 &

To execute a command entered in the field, click on the **Execute** button.

Command File

This field is used to specify a startup file you want to run on a host or edit it.

When you click on the scroll arrow beside the **Command File** field, a drop-down box will display all startup files stored in the XLitePro's home directory.

In order to execute instructions in a startup file, select from the box or enter into the field a name for the file you want to run and press the **Execute** button. If the file exists, its first line appears in the **Command** field.

Another way of specifying a startup file is to enter into the **Command** field a file name preceded by the **@** sign, for example **@mystartup.su**.

Edit

When you press on this button, the Microsoft Notepad is invoked and the startup file specified in the **Command File** field will be opened. This lets you view or modify the file before running it.

Log File

When you enable this option, whenever you execute a single command or a startup file, any startup or error message information that the host or client would normally display will be sent to a file. The log file resides in the XLitePro's home directory. The file's name differs depending on how you

execute commands.

- If you execute a single command entered into the **Command** field, all log information will be saved in a file called **startup.log**.
- If you run a startup file, the log file will be saved as *filename***.LOG**, where *filename* is the name of the startup file.

Information will be sent to a log file for the amount of time specified in the **Exit after Timeout of** field, after which the socket will be closed.

Wait for RLogin Password

This option is used if you want to enter your password only when a prompt from the RLOGIN daemon appears.

Exit after Timeout of

This field defines the amount of time the socket will stay open after the host connection has been made. The default value is of ten seconds. If the connection gets established but the invoked program does not get time to start up, you can set this to a higher value.

empty Default UserID for Jobs

If selected, this check box suppresses using the MS Windows logon name as default in the Jobs' "Startup Login" dialogs.

Default

This button will initialize all parameters to their default values.

Status Messages

Whenever you execute a single command or a startup file, any startup or error message information that the host or client would normally display is sent to the box.

Startup Jobs

A job is a named task with a certain set of parameters. Job's main parameters include: start method, login information, command lines, and settings.

The **Startup Job Name** list contains names of currently available jobs. You can create an icon for a job and choose a Program Group to contain these icons. By default, this is the **StartIcons** folder.

A job may be launched by double-clicking on its icon (in the **StartIcons** folder by default). You can have up to 512 jobs.

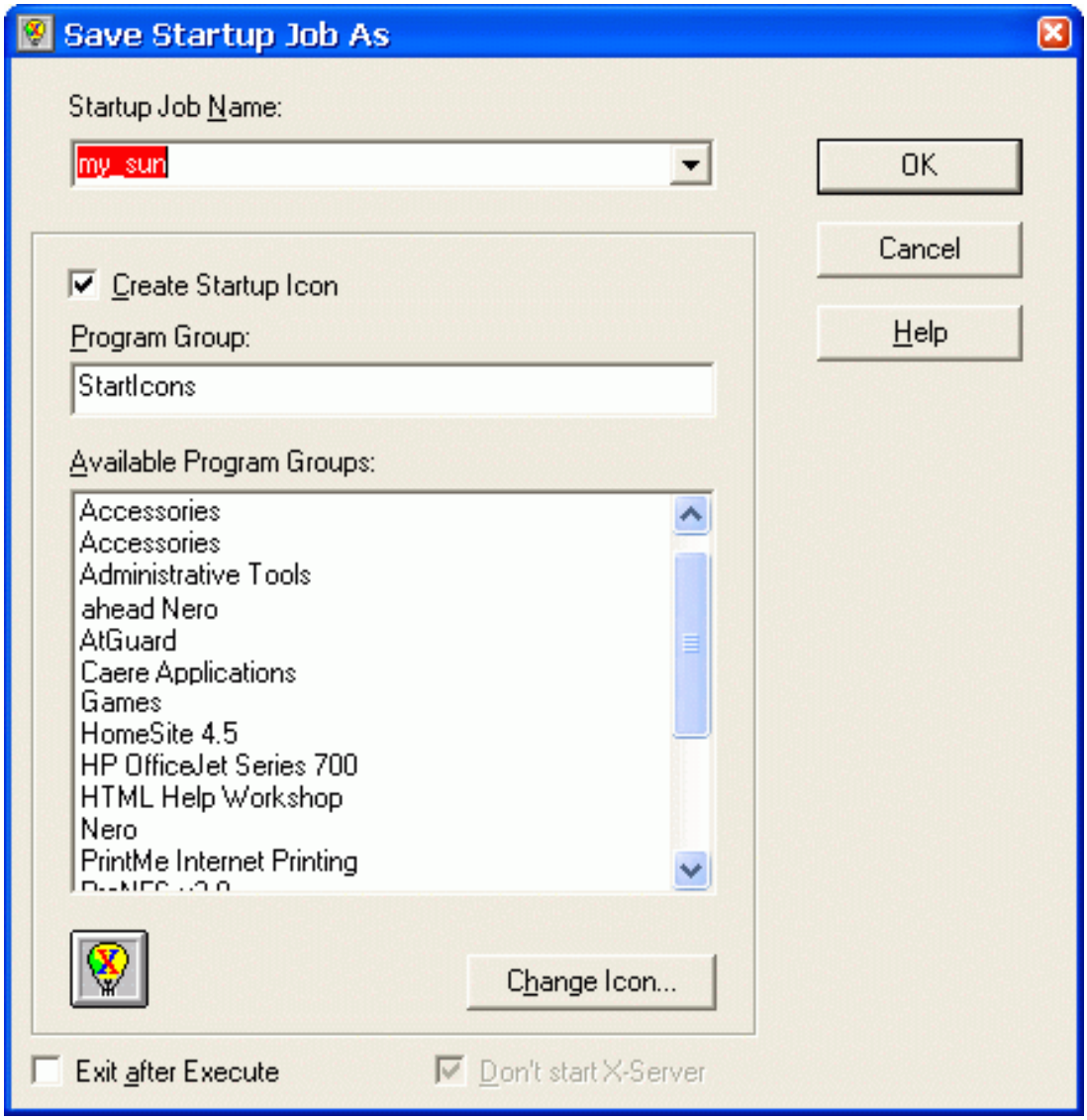
The **Save Job As**, **Open Job** and **Remove Job** buttons in the **Startup** window are for manipulating with Startup jobs.

The **Save Job As** button creates a job and adds its name to the list. The **Open Job** button opens a specified job for modifying. You may remove an opened job by pressing the **Remove Job** button.

Any changes of the **Startup Job Name** list are automatically displayed in the corresponding Program Group (**StartIcons** by default).

Saving a Job

When you click on the **Save Job As** button, it displays the following dialog box:



This dialog box allows you to create a job, i.e. to store current parameters under a certain name. If a job with the same name already exists, it will be overwritten; if it does not exist, the new job will be created and added to the **Startup Job Name** list. The default icon for the job may be changed with the **Change Icon** button.

Startup Job Name

This field defines a name for the job to be saved. In the drop-down box, you can view all available job names in the list.

Create Startup Icon

If this check box is checked, then a job will be displayed in the Program Group you will choose for your jobs (**StartIcons** by default) with an icon you will specify; otherwise the job will be only added to the **Startup Job Name** list.

Program Group

This field is used to specify a group to contain your job's icons (**StartIcons** by default). You can enter a name in the field or select a name from the **Available Program Groups** field. The Program Group will first be cleared and then all jobs from the **Startup Job Name** list (marked to **Create Startup Icon**) will be moved into it.

Available Program Groups

This box lets you select a group to contain your job's icons. The name will appear in the **Program Group** field.

Exit after Execute

If this check box is disabled, then the **Startup** dialog box will stay on after executing the job and you will be able to continue working on it.

Don't start XServer

If this check box is disabled, then XServer will be started up (if it is not running yet) when opening the job (or launching it by double-clicking on its icon) and before executing its commands.

Otherwise, you may open the job and execute commands which do not launch X clients and so do not require XServer (if it is not running yet).

Note that if you launch a job by double-clicking on its icon, then the job will execute in batch mode.

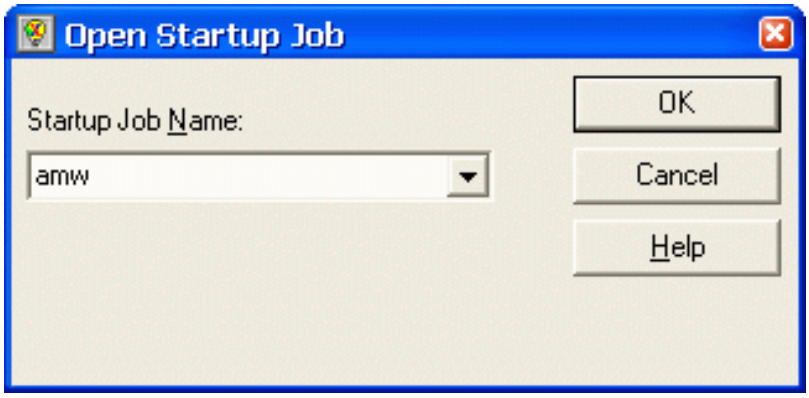
Change Icon

When you click on the button, it displays the following dialog box:

This box gives you a possibility to choose a suitable icon for the job to be saved. The **File Name** field displays a current file name and the corresponding **Icon List**. If you want to assign to your job an icon other than by **Default**, select the desired icon from the current **Icon List**. To change list of available icons, press the **Browse** button and choose a suitable icon from the **Open** window.

Opening a Job

When you click on the **Open Job** button, it displays the following dialog box:



This box allows you to view the **Startup Job Name** list and to select a certain job from it. After highlighting a job in the drop-down box or entering a job name into the edit field, press **OK** to open the job. Its parameters will be displayed in corresponding fields of the **Startup** window, so you can view or change them before running the job.

6. Configuring XLitePro **< previous | content | next >**

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

Running Startup with Command Line Parameters

The 'xini' parameter

You can launch Startup with the 'xini' command line parameter:

PATH\xstartup.exe -xini <IniFilePath>

where <IniFilePath> specifies a full path to a specific ini-file and PATH indicates your XLitePro home directory.

This feature allows you to run several Startup sessions each with its own ini-file (i.e. settings).

In order to do so, you can create a new Startup shortcut (e.g., in the XLitePro Programs' folder) and fill in the **Target** field in Properties of it with the command. For example, the **Target** field in Properties of the Startup shortcut might contain the following line:

"PATH\xstartup.exe" -xini myxwp1.ini

By default, the field contains a call of Startup with no arguments, and the **xwp.ini** file will be used in this case.

To create your specific ini-file, you can copy the **xwp.ini** file and then change required parameters with the Startup utility by starting it with the command line parameter.

The 'lxdn' parameter

The "-lxdn [<DisplayNumber>]" command line parameter is used to set up the "$(DISPLAYNUMBER)" macros (for the DISPLAY session environment variable if a remote daemon supports it). This provides correct X11-forwarding (if set).

The "-lxdn" option means looking up running XServer (i.e., Startup understands the local XServer's "DynamicDisplayNumber" mode and correctly looks up running XServer). In this case, XServer must already have been started.

The 'mout' parameter

The "-mout [<outputfile>]" command line option is used to provide a "readable" text-tracing to the "outputfile" log-file for your session.

The 'log' parameter

The "-log <LogLevel>" option is used to set up the level of tracing (0..7).

The "[Startup]" Section of the ini-file

The "[Startup]" section of the ini-file may have the following entry lines you can customize for particular needs and applications.

ExitAfterExecute=1
PromptPassword=1

to specify the auto-execution mode of the default command line.

Starting XServer

You can start XServer by double-clicking on the **XSession** icon from the XLitePro Programs' folder.

XServer will be started in a default window mode (according to settings stored in the ini-file used). When XServer is running, you can change the current window mode by selecting an appropriate item from the XServer's Control menu or by launching the XSettings utility. (See section **Using XSettings** in Chapter **Configuring XLitePro**.)

You can launch XSettings either from the XLitePro Programs' folder or from the XServer's Run menu. If you save new settings from XSettings in the ini-file, then you must restart XServer to re-read the file and to take them effect.

Also, see section **Using ComSetup** in Chapter **Configuring XLitePro** on how to make communication settings relating to the networking aspects of XLitePro that operate with the TCP/IP transport interface.

There are some additional settings you can specify to customize your X-sessions. They are described in section **The "[XSETUP]" Section of the ini-file** below.

Initiating Remote Login Sessions

Working with XServer typically includes the following steps:

1. Initiating a remote login session;
2. Starting a session manager. The first client run in a login session acts as the session manager. A useful session manager is 'xterm' or a window manager, both of which allow other clients to be run;
3. Running clients;
4. Completing the session with the session manager.

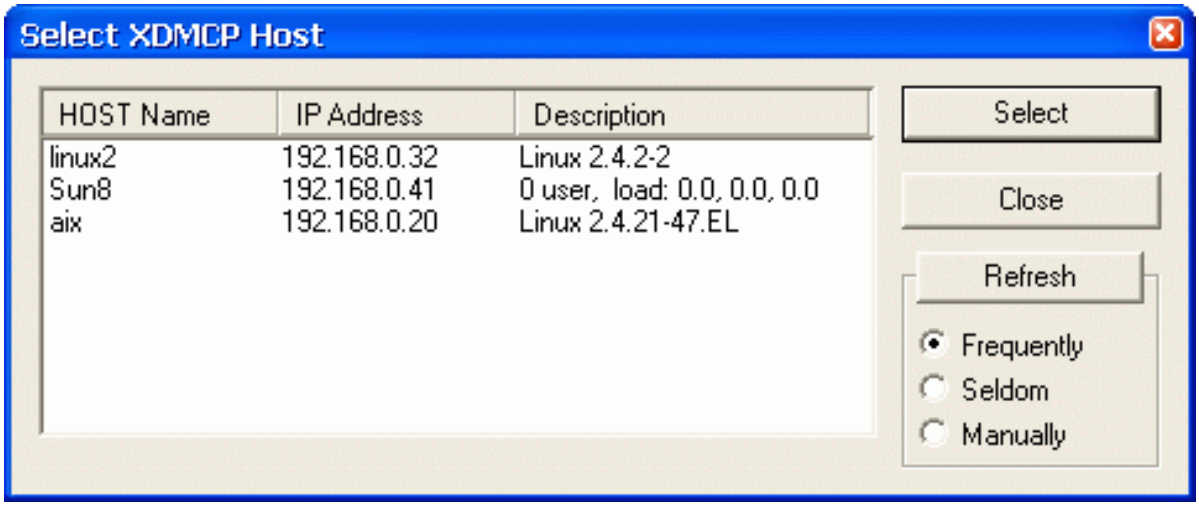
First of all a user should initiate a login session by using Telnet, XDMCP, or Startup (REXEC, RSH, RLOGIN) methods.

Note, a special case of working with clients is their starting from other X terminals without initiating remote login session on your PC. Clients have to be started with the option **-display** that defines your PC as a display.

The XDMCP Method

To use XDMCP, you have to run XSettings and preset the **Use XDMCP** check box to the enable state. You can use the XDMCP method only if you are sure that the host you want to connect to supports XDMCP and settings for XDMCP are correct there. (See section **XDMCP Settings** in Chapter **Configuring XLitePro**.)

If you select the **Broadcast** XDMCP mode in the **XDMCP Settings** of XSettings, then the **Select XDMCP Host** window will appear after loading XServer. In this window, you will see the XDMCP hosts running on your network, and you can select one of them to start an X-session.



If you want to use the **Query** or **Indirect** XDMCP mode, you should specify the network node name or IP address for the host you want to connect to in the **Connect Host** field in the **XDMCP Settings** of XSettings.

After establishing the X connection, XDMCP contacts a 'xdm' process running on a host system. Then 'xdm' initiates login session as follows:

1. An optional authentication procedure takes place in which 'xdm' proves to the terminal that it is authorized to manage that display.
2. A session key is generated to be used for subsequent client authorization.
3. A username/password is requested to authenticate the user.
4. A session manager is run, typically **/usr/lib/X11/xdm/Xsession**, which is typically a shell script that ends up executing the required session manager such as 'xterm'.

When the session manager terminates, all other clients have their windows destroyed and should then terminate themselves. A new session is then initiated by 'xdm'.

Many XDMCP parameters may be redefined if required in **/usr/lib/X11/xdm/xdm-config** and **/usr/lib/X11/xdm/Xresources**. See manual xdm(1) for details. In particular, sites may find it useful to tailor the login window greeting in Xresources to identify the host system:

xlogin*Login*namePrompt: CLIENTHOST login:

CLIENTHOST is replaced by the host name of the system running 'xdm'.

The Telnet Method

Although XDMCP is recommended for convenience, it may sometimes be necessary to start clients without it, such as when the host system software does not support XDMCP or does not have it configured correctly. In these cases, you can run clients by typing shell commands in the normal manner. To gain access to a shell, you must first log in to the host system, for example on a dumb terminal such as the system console. For convenience, XLitePro has the Telnet virtual terminal network interface. Start the Telnet program, connect to a host and enter commands to start clients. (See Chapter **Telnet**.)

Note: the Telnet and XServer programs use the same Keyboard Definition file you specified.

REXEC, RSH and RLOGIN Methods

The REXEC, RSH and RLOGIN methods can be used for automating host access and X client startup with using the REXEC, RSH and RLOGIN protocols respectively. You can use the Startup program to enter a single command line and execute one on a host, or run a local startup file that will automatically start one or more X clients. If XServer has not been running, it will start.

In order to run Startup, your host system must support the REXEC, RSH or RLOGIN protocol. REXEC operations may be done either by direct access to a remote host or through one of the established SSH1/SSH2 connections. (See Chapter **Startup**.)

Note: if you use Startup to start your XServer, and XDMCP is enabled, the program will switch off

the XDMCP startup method. This will be done because the host XDM script will most likely run before your command or file.

The REXEC, RSH and RLOGIN session dialogs are quite similar. Each of them asks for a host name, user ID name, and command. In addition, REXEC needs to know a user password.

For RSH method, some hosts require your PC to be authorized before using RSH, as it does not require a password. On UNIX systems, you need to add your PC name to the **/etc/hosts.equiv** and **$HOME/.rhosts** files on your host.

Also, check the **/etc/inetd.conf** and **/etc/services** files on the host. The **services** file is a local source of information regarding each service available through the Internet. The **services** file must contain an entry for each service. The **inetd.conf** file contains a list of servers that 'inetd' invokes when it receives an Internet request over a socket.

Terminating XServer

To terminate XServer, click on the **XSession** icon or a box at the top left corner of the X-session window and select the **Close** command from the Control Menu. Then all X client sessions will be terminated. To terminate individual X clients, use the client's standard terminate program command.

Note, XServer is also terminated by terminating Microsoft Windows.

A Note on Copy and Paste

A number of copy and paste commands described in this chapter resemble an X Selection. In the X environment, the X Selection is a buffer with a specific name and properties that stores data you select in an X client's window.

The X Selection also serves as the source for certain XLitePro copy commands and the target for certain paste commands.

X Selections are the mechanisms you can use to copy data from X to Windows or vice versa.

The name of the X Selection a client uses and how it is used (for text or image) depends on the client. Most use the **PRIMARY** X Selection, but other selections like **SECONDARY** and **CUT_BUFFER0** through **CUT_BUFFER7** are also defined. Some clients can make use of more than one X Selection.

If you find that data is not being copied or pasted, XLitePro provides the **X Selection** item on the **Options** menu. This item lets you define which X Selection to use (see **The Options Menu** below).

The **Auto Clipboard Copy&Paste** check box (in the XSettings window) controls the mode of connection between the MS Windows Clipboard and the current X Selection. When enabled, then changing of the MS Windows Clipboard contents will force the same change in the current X Selection. Also, changing of the contents of the current X Selection (from the active application window) will automatically cause copying it into the MS Windows Clipboard (so you are ready to paste).

Also, you can control the mode of connection between MS Windows Clipboard and a current X Selection by specifying the **ConnectXWClipboards** setting described in section **The "[XSETUP]" Section of the ini-file** below.

Also, see **Appendix C Troubleshooting** for examples.

Single Window Mode

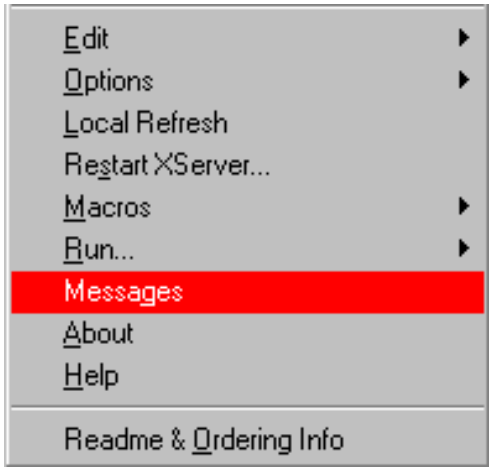
This mode presents all X clients in a single X-session window. Within the window, the window management and all other functions are typically controlled by an X Window System manager you start on a host (e.g., olwm, CDE, twm, mwm, etc.).

If you want to use a remote window manager, you should use the XDMCP method or start a remote window manager from the UNIX host (e.g., start xterm and then start the required window manager from the xterm window).

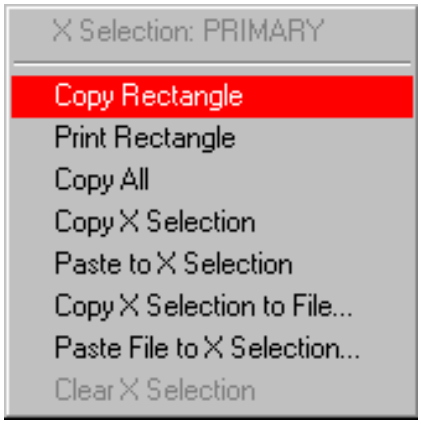
The X-session window itself can be sized and moved like any Microsoft Windows window. You can use scroll bars to display all parts of the virtual screen. Pressing a scroll arrow will move the window a pixel at a time. Dragging the scroll elevator or clicking on either side of the elevator will move the window more rapidly.

Note: the local mwm program (based on OpenMotif 2.0) may be useful in this mode to provide windows managing functions (moving, resizing, etc.). This allows not using a remote window manager. You can launch the local mwm program from the X-session's Run menu.

Functions that affect XServer as a whole can be selected from the X-session's Control Menu. This menu will be presented when you click on a box at the top left corner of the X-session window.



The **Edit** menu gives you access to all XLitePro Copy & Paste commands. A number of these commands specify the current X Selection as the source or target of the command (see **A Note on Copy and Paste** earlier in this chapter).



Example of Initiating a Login Session by Using REXEC to Start xterm

First, make sure that your remote host is accessible from your PC (e.g., with the Ping utility) and supports the REXEC protocol. Also, make sure that your PC is accessible from your remote host. If OK, then you may initiate a login session by using REXEC like the following steps:

1) Specifying required settings

- Start XSettings by double-clicking on the **XSettings** icon from the XLitePro Programs' folder (the XSettings window should appear)
- Select **Single** from the Window Mode List
- Disable the **Use XDMCP** check box from the XDMCP section
- Specify other required settings
- Click **OK** to save your settings and exit XSettings

2) Initiating X-session

- Start XServer by double-clicking on the **XSession** icon from the XLitePro Programs' folder (the empty X-session window should appear)

3) Initiating the Startup session

- Start Startup by double-clicking on the **Startup** icon from the XLitePro Programs' folder (or from the Run menu of XServer); the Startup window should appear
- Choose **REXEC** in the Start Method section
- Enter the network name or IP address for the remote host you want to connect to in the **Host Name / IP Address** field in the Login section
- Enter your User Name in the **User ID** field
- Enter your **Password** required for your login to the remote system
- In the **Command** field, enter the following line

  `/usr/X11R6/bin/xterm -display $(LOCALIP):0`

- Specify other required settings

- Click the **Execute** button

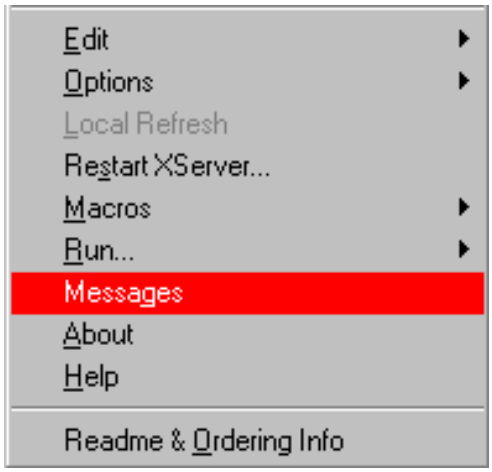
Note: The path to xterm depends on the operating system's configuration used on the host (e.g., the path may be **/usr/bin/X11/xterm**).

The xterm window should appear on your display. You can then interact with the host by typing commands in the window and starting remote applications (e.g., xclock).

Full Screen Mode

This mode presents all X clients in a single root window taking up full the screen outside the Microsoft Windows graphical environment. The window management and all other functions are typically controlled by an X Window System window manager you start on a host.

For transition from the X Window System graphical environment into the Microsoft Windows graphical environment, the **ScrollLock** key is used. In the MS Windows environment, the **Full Screen** mode displays the **XSession** icon at the bottom of your display. Functions that affect XServer as a whole are controlled by clicking this icon.



For transition from the Microsoft Windows graphical environment into the X Window System graphical environment, the **Alt+Esc** or **Alt+Tab** keys are used.

If you used the Virtual Root box (described in section **Using XSettings** in Chapter **Configuring XLitePro**) to define a virtual screen larger than your display, you can display all parts of the virtual screen by moving a cursor to borders of PC's physical screen. Trying to move the cursor beyond a physical screen border results in the virtual screen moving in an opposite direction.

Example of Initiating a Login Session by Using XDMCP

First, make sure that xdm-daemon is running on your remote host and check the XDMCP configuration. Make sure that your remote host is accessible from your PC (e.g., with the Ping utility). Also, make sure that your PC is accessible from your remote host. If OK, then you may initiate a login session by using XDMCP like the following steps:

1) Specifying required settings

- Start XSettings by double-clicking on the **XSettings** icon from the XLitePro Programs' folder (the XSettings window should appear)
- Select **Full Screen** from the Window Mode List

- Enable the **Use XDMCP** check box and click **Settings** from the XDMCP section (the XDMCP Settings window should appear)
- Enter the IP address or the network name for the remote host you want to connect to in the **Connect Host** field
- Choose **Query** in the XDMCP mode section to specify requesting to the host
- Specify other required XDMCP settings
- Click **OK** to save your settings and exit XDMCP Settings
- Specify other required settings
- Click **OK** to save your settings and exit XSettings

2) Initiating X-session

- Start XServer by double-clicking on the **XSession** icon from the XLitePro Programs' folder (the X-session window should appear with the Login window of XDMCP)

9. Using XServer **< previous | content | next >**

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

Multiple Window Mode

This mode provides the most integrated way to run Microsoft Windows and X on the same computer. When an X client starts, it appears in a window like any other displayed by MS Windows. Each client you start creates its own window on your display. You can have up to 2048 different top-level windows.

The client window's controls (i.e. its borders, the Control Menu box, move window functions, etc.) are all handled by Microsoft Windows on your PC. This makes for an integrated environment where you can manipulate X client windows in the same way as windows displayed by Explorer or any other MS Windows' program.

In the **Multiple** window mode, Microsoft Windows works as a local window manager for your X clients. Since window management functions do not have to be provided from a remote host, this mode speeds up window management functions and reduces network traffic. In this mode, XServer can work on multi-monitors PCs.

Functions that affect XServer as a whole are controlled by clicking the **X-session** icon.

Note: the XSession icon cannot be changed into a window in the **Multiple** window mode, as its only purpose is to control the overall settings for X clients that appear in their own windows.

Note: the MS Windows window manager is locally used in the **Multiple** window mode and any other window manager is blocked. Some remote windows managers close the X-session if they detect any other running window manager (e.g., the XDMCP method can fail because of this). In this case, you cannot use the **Multiple** window mode. You can use it if you do not want to use a remote window manager.

Example of Initiating a Login Session by Using Telnet to Start xterm

First, make sure that your remote host is accessible from your PC (e.g., with the Ping utility). Also, make sure that your PC is accessible from your remote host. If OK, then you may initiate a login session by using Telnet like the following steps:

1) Specifying required settings

- Start XSettings by double-clicking on the **XSettings** icon from the XLitePro Programs' folder (the XSettings window should appear)
- Select **Multiple** from the Window Mode List
- Disable the **Use XDMCP** check box from the XDMCP section
- Specify other required settings
- Click **OK** to save your settings and exit XSettings

2) Initiating X-session

- Start XServer by double-clicking on the **XSession** icon from the XLitePro Programs' folder (the empty X-session window should appear)

3) Initiating the Telnet session

- Start Telnet by double-clicking on the **Telnet** icon from the XLitePro Programs' folder (or from the Run menu of XServer); the Telnet Connect Host window should appear
- Enter the network name or IP address for the remote host you want to connect to
- Change the default Telnet port number if required
- Click **OK** to start the Telnet session with the remote host (the Telnet window should appear)
- Specify the login information required for your host system (e.g., User Name and Password)

4) Starting the xterm program from the Telnet window

- Run the following command line:

e.g.,

/usr/bin/X11/xterm -display IP_address_of_your_PC:0&

or

/usr/bin/X11/xterm -display Name_of_your_PC:0&

(in the latter case, you should have Name_of_your_PC and IP_address_of_your_PC in the **/etc/hosts** file on your UNIX host).

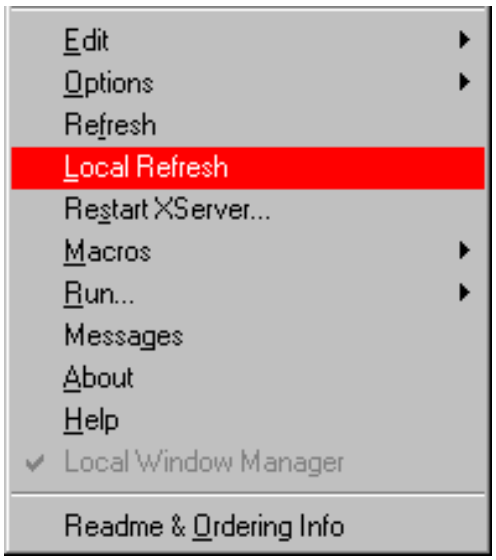
The xterm window should appear on your display. You can then interact with the host by choosing commands from displayed menus, or by typing commands in the window and starting remote applications (e.g., xclock).

Note: The path to xterm depends on the operating system's configuration used on the host (e.g., the path may be **/usr/X11R6/bin/xterm**).

Note: You should not start a remote window manager after that.

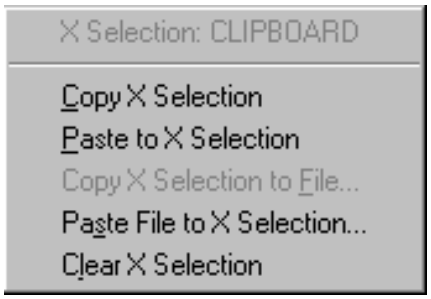
The X Client's Menu

On the X client window's Control Menu, you can use both MS Windows commands and menu items added by XLitePro.



Restore, **Move**, **Size**, **Minimize**, **Maximize** and **Close** all function as with any MS Windows application.

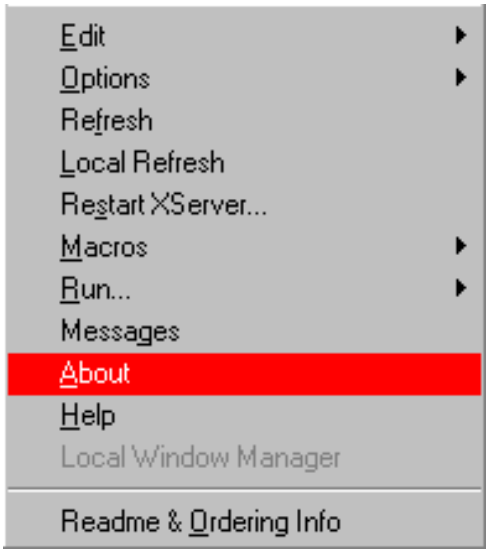
The **Edit** menu gives you access to all XLitePro Copy & Paste commands. A number of these commands specify the current X Selection as the source or target of the command (see **A Note on Copy and Paste** earlier in this chapter).



The **Edit** menu presents the same set of commands displayed from the server's Control Menu in the **Single** window mode (see section **The Edit Menu** of **XServer Commands** below), but the **Copy Rectangle**, **Print Rectangle**, and **Copy All** commands affect only the current client's window.

Multiple+RemoteWM Mode

This mode is the above **Multiple** mode, but the local MS Windows window manager does not control windows of X clients and a user has to run any suitable remote window manager. The mode is very convenient when users use CDE-like interface where a remote window manager provides its own tool/ task bar. In this mode, XServer can work on multi-monitors PCs.



Note that the **Restore**/**Minimize** items on the XServer menu provide "visibility" of XServer background in an additional small window. This "x-root window" may be used to control functions of the remote window manager used (usually by pressing mouse buttons).

XServer Commands

In the **Multiple** and **Full Screen** modes, XServer commands are displayed by clicking on the X-session icon.

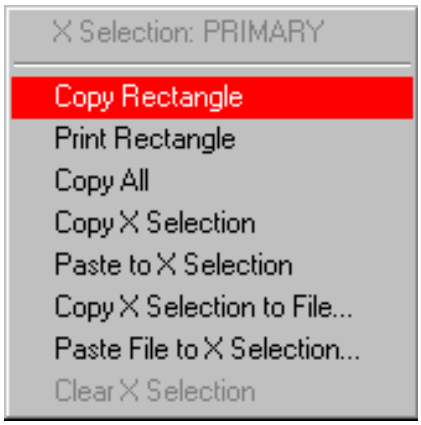
In the **Single** mode, XServer commands can be accessed by clicking on the Control Menu box at the top left corner of the X-session window.

The top of the Control Menu presents the standard MS Windows commands (**Restore**, **Move**, **Size**, **Minimize**, **Maximize**, and **Close**).

The bottom of the menu presents XServer commands. They are described below.

The Edit Menu

When you select **Edit**, the Edit Menu commands are displayed:



All available Edit commands are described below.

The first item shows you the current X Selection.

Note: Whenever X Selection is the source or target of a copy or paste command, you can define the X Selection being used with the **Options-X Selection** menu item. This item is described in **A Note on Copy and Paste** and **The Options Menu** sections in this chapter.

Copy Rectangle

This command is available in the **Single** mode only. It copies a rectilinear area you select on any visible part of the X-session window (i.e. a pixel image) to the MS Windows Clipboard (regardless of the **ConnectXWClipboards** setting). When you choose this item, a camera icon is displayed. Click (and hold down) the left mouse button at the rectangle's origin and drag the mouse to define the

rectangle. As you drag the mouse, lines will delineate the rectangle to be copied. When you release the left mouse button, the selected rectangle will be copied to the MS Windows Clipboard. You can then save and paste this image from the MS Windows Clipboard into programs running under MS Windows that support bitmap paste.

Print Rectangle

This command is available in the **Single** mode only. It prints a rectilinear area you select on any visible part of the X-session window (i.e. a pixel image). When you choose this item, a camera icon is displayed. Click (and hold down) the left mouse button at the rectangle's origin and drag the mouse to define the rectangle. As you drag the mouse, lines will delineate the rectangle to be printed. When you release the left mouse button, the selected rectangle will be printed (after you will select your target printer).

Copy All

This command is available in the **Single** mode only. It copies all visible parts of the X-session window (i.e. a pixel image) to the MS Windows Clipboard (regardless of the **ConnectXWClipboards** setting). You can then save and paste this image from the MS Windows Clipboard into programs running under MS Windows that support bitmap paste.

Copy X Selection

This command copies text and graphics from the current X Selection to the MS Windows Clipboard (regardless of the **ConnectXWClipboards** setting). You can use an X client's selection procedure to select the data you want to copy. Once data is in the Clipboard, you can use the Clipboard to save data or paste it into any MS Windows application that supports copy and paste.

Paste to X Selection

When you choose this command, text or graphics that is in the Microsoft Windows Clipboard is pasted to the current X Selection (regardless of the **ConnectXWClipboards** setting). You can then use an X client's paste procedure to paste the data into an X client.

This function works for both text and graphics, although many clients do not support graphics.

Copy X Selection to File

This item lets you save any text that is in the current X Selection to a file on your PC. You can use an X client's selection procedure to select the data you want to copy. When you select the item, you will be prompted for the name of the file you want to save the data to. When you click OK, the current X Selection will be copied to the specified file.

Paste File to X Selection

This item lets you paste any text file that is on your PC to the current X Selection. When you select the item, you will be prompted for the name of the file you want to paste. When you click OK, the specified file will be pasted to the current X Selection. You can then use an X client's paste procedure

to paste the X Selection into an X client.

Clear X Selection

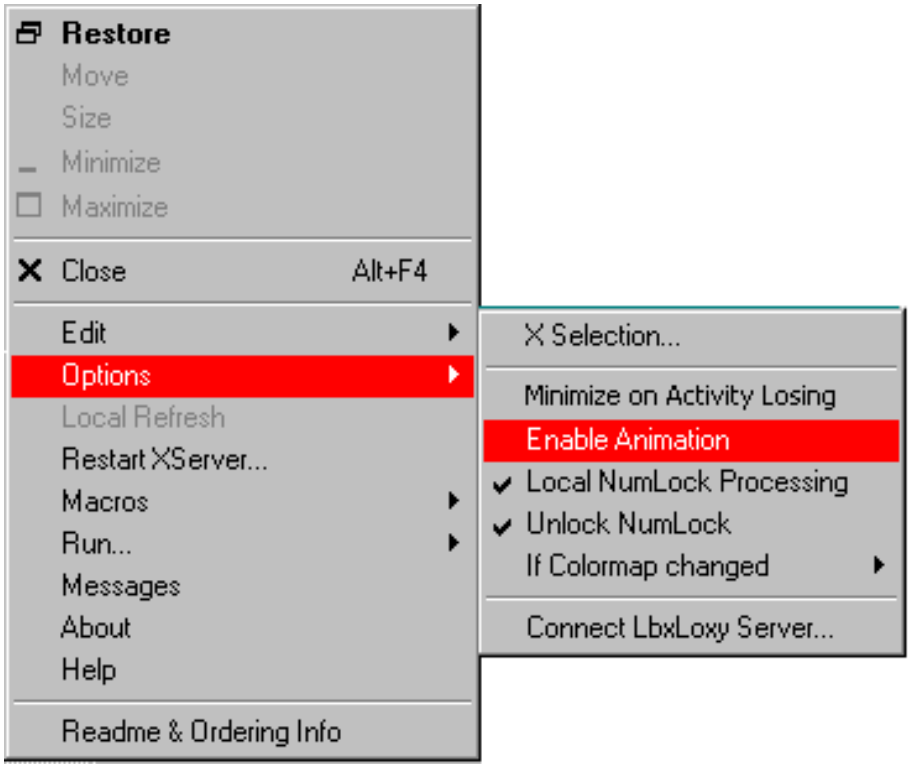
This item will be grayed unless XServer has control of the X Selection.

Whenever you choose the **Paste to X Selection** or **Paste File to X Selection** commands, XServer takes control of the X Selection. You can use the **Clear X Selection** function to clear the buffer and free the memory required to maintain it.

Note: This function can also be performed by making another selection in an X client application.

The Options Menu

The Options menu controls options that affect the operation of XServer as a whole. When you select **Options**, the Options Menu commands are displayed:

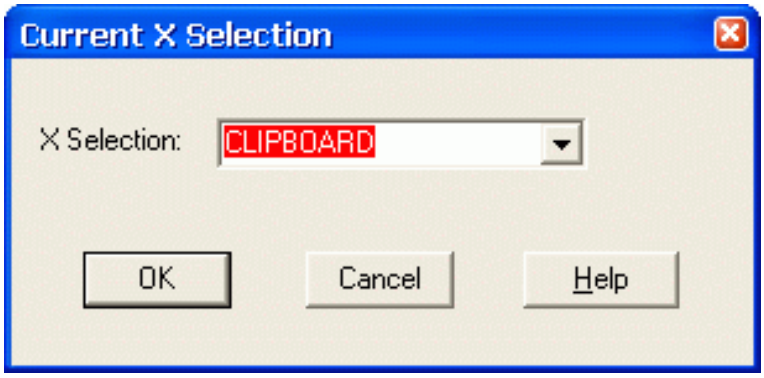


All available Options commands are described below.

X Selection

This option determines an X Selection that will be used by the XServer's **Copy X Selection**, **Paste to X Selection**, **Copy X Selection to File**, **Paste File to X Selection** and **Clear X Selection** commands of the Edit menu.

Whenever you select this option, a dialog box appears on your display prompting you to specify which X Selection you want to used. The current choice will appear at the top of the Edit menu. Use the mouse to select the choice you want from the list box and then click OK.



Most X applications use PRIMARY. For X clients that support SECONDARY, CLIPBOARD, or CUT_BUFFER0 through CUT_BUFFER7, you can change the current X Selection by selecting it from the list.

Note: you can also use the edit field at the top of the dialog box to enter a name for a custom X Selection. This is only useful if the name is defined by an X client. If you do this, the name will be saved and will appear in the dialog box whenever you select this option.

Minimize on Activity Losing

This option is only available in the **Full Screen** mode. If you select this option, the XServer's window will be iconified each time the focus changes to another window. Otherwise, it can be obscured by other windows.

Enable Animation

The **Enable Animation** mode provides careful operations with colors while drawing. Note that this mode decreases performance of XServer.

Local NumLock Processing

If this option is enabled, then XServer will process numeric keypad keys, depending on the (local) NumLock key state, and results will be sent to X clients. Otherwise, X clients process the keys (remotely).

Unlock NumLock

This option corresponds to the **Unlatched NumLock** check box in the XSettings window. If this option is enabled, XServer will consider the NumLock key as a normal key (non-toggling). The NumLock key is **unlocked** by default. This option was implemented to suppress the NumLock state's influence on some X-window managers and programs.

If Colormap Changed

The **If Colormap Changed** menu lets you specify XServer response to the hardware colormap (Windows system palette) changes. An X client can require installing its own colormap into the hardware colormap when the user indicates that the application should be active. When this occurs, all other applications can appear in false colors. As soon as you change input focus to another X

client, you expect to see right colors displayed in its windows as well as windows using the same colormap as the X client.

XServer regenerates colors of windows using a policy specified by one of the following options:

Redraw

This option causes redrawing the active X client windows as well as windows using the same colormap as the X client.

Update Colors

If you choose this option, XServer will directly update the colors in all the X client windows when you activate an application that requires changing the hardware colormap. The option provides a reasonable compromise between performance and color accuracy.

Note that this option is not supported by some video drivers.

If you do not check any of these options, XServer will do nothing to regenerate colors of windows when changing the hardware colormap. The colors displayed in all the X client windows will potentially be incorrect until the windows update their display. You can redraw all X client windows by choosing the **Refresh** or **Local Refresh** item from the XServer's Control Menu.

Move Window to (0,0)

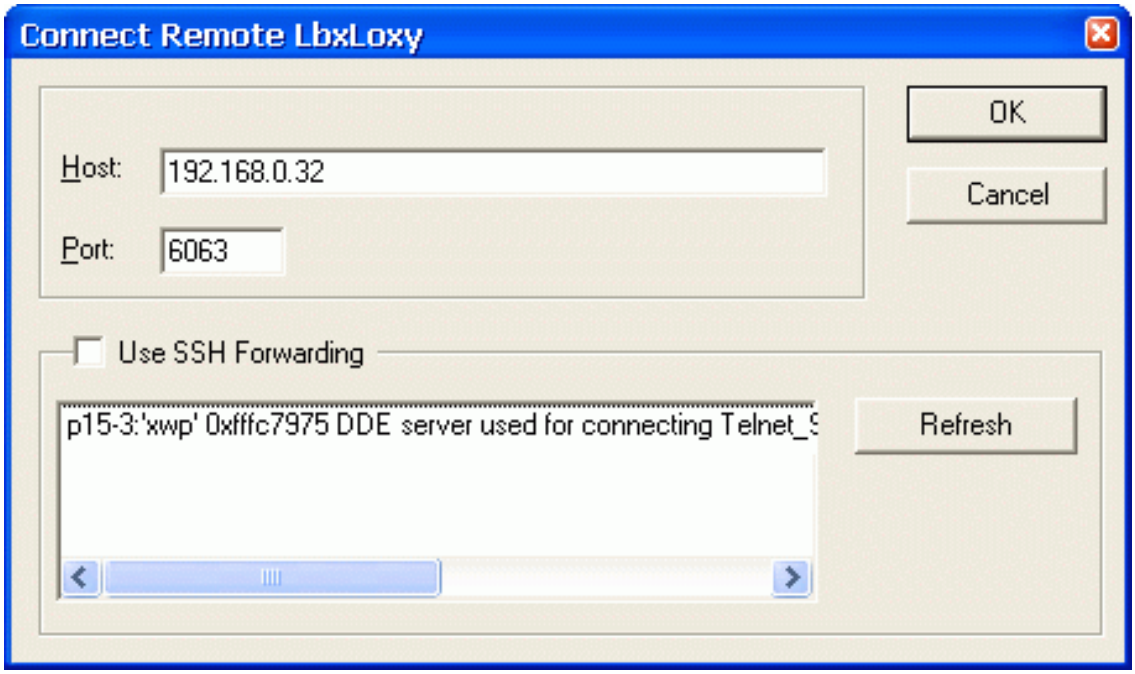
Implemented in the XServer/Multiple Control menu, this option allows users to move any window to the upper left corner of the Windows screen. This feature is useful to view an invisible window when it is out of the visible screen.

Connect LbxLoxy Server

The LbxLoxy Server is a Low BandWidth X proxy derived from sources of the lbxproxy (included in XFree 4.10). In addition to its original capabilities, LbxLoxy can wait for connection to a remote X server instead of initiating the connection itself. Thus, when external connections to XServer are prohibited (e.g. by a firewalling proxy) XServer can initiate a connection to a listening socket for LbxLoxy.

Connection to the running remote LbxLoxy Server may be done through either the direct access insecure channel mode or the secure (SSH) channel mode. See section **Using LBX** below on how to use Low Bandwidth X (LBX) network-transparent protocol for running X Window System applications.

When you click the **Connect LbxLoxy Server** item, the **Connect Remote LbxLoxy** dialog box will appear.



You can see a list of actual SSH1/SSH2 connections you have previously established (for example, by the SSH1 or SSH2 Client of the Telnet_SSH program). The **Refresh** button allows you to renovate the list.

If you want LBX operations to be done in secure connection mode, select a desired remote host (an active SSH server) from within the list and enable the **Use SSH Forwarding** check box. In the SSH1/SSH2 mode, the "Dynamic Port Forwarding" feature of the Telnet_SSH/SSH1 or Telnet_SSH/SSH2 Client and XServer will automatically be used, so you need not set up any port forwarding manually.

If you disable the Use **SSH Forwarding** check box then the direct access insecure mode will only be used for LBX operations.

If the LBX server and SSH server you are using in the X-session are on the same remote host then the **Host** field can contain "localhost" or "127.0.0.1". This is more preferable than its actual IP address in most cases (but depends on how the host has been configured).

In the **Port** field, you specify a port number that will be listening for connection from XServer (default is 6163).

Open a login session on a remote host.

Run lbxloxy on the remote host with option '-waitXServer[:port]' where port is a port number that will be listening for connection from XServer (default is 6163):

$ ./lbxloxy -waitXServer:6163 &

or (default)

$ ./lbxloxy -waitXServer &

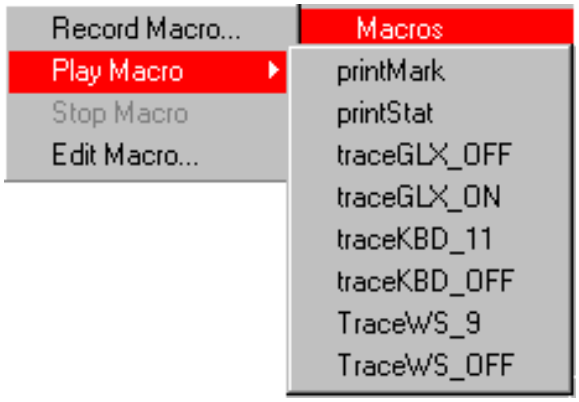
When connection is established, lbxloxy will open a display port for listening (default is 6163). For convenience, you may set the DISPLAY environment variable:

$ export DISPLAY=:63.0

After that, you are able to run your remote X application(s).

The Macros Menu

This menu lets you create macro-files to remember sequences of your keystrokes. A sequence of remembered keystrokes is called a keystroke macro. Then, whenever you want, you can play back these keystroke macros in a target application window by choosing an appropriate macro-file. Keystrokes are remembered in macro-files as ASCII strings in a special syntax, so you can use text editors (e.g., Notepad) to edit the files. By default, macro-files have the **.mac** extension and reside in the package's home directory.



Record Macro

When you click on this item the standard Save As dialog box named as **File for Macro Recording** appears. You can specify a macro-file name you want to record macro to. Pressing **Save** opens the specified file for writing and turns on the record macro mode. Then, in the target application window (e.g., Terminal), you perform the keyboard actions you want remembered. The mode is effective until you click on the **Stop Macro** item on the Macros menu or in the **Macro recording is running** information box, which indicates that the mode is on. Before using the **Record Macro** option get your target application to the point where you are ready to type the first keystroke of the macro.

Play Macro

This option lets you select a macro-file you want to play back. When you select a file from the macro-file list it is played back immediately in the active X client's window (with actions depending on the application type, e.g., Terminal or Text Editor). If a macro was created for a particular application, you can play it back whenever you are working in that application. Before using the **Play Macro** option get your target application to the point where you want to play back the macro.

Stop Macro

You use this option when you have finished the last keystroke of the action you want to record. This option turns off the record macro mode and closes the current macro-file. Now you can play back the macro-file just created or any other available one.

Edit Macro

When you click on this item the standard Open dialog box named as **Macro File to Edit** appears. You can choose a macro-file you want to edit. Pressing **Open** invokes Notepad for you to edit the file specified.

Notes

1. It is not recommended to edit macro-files with Notepad.
2. Nested macro-files are not allowed.
3. One way you can take advantage of macro-files is substituting a keysym for a key (defined in a keyboard definition file) with a keystroke sequence (defined in a macro-file). Appendix A, **Keyboard Mapping File Format**, describes normal keysym statement syntax. The extension to this is that a keysym can have the form of prefix **MAC_** followed by a macro-file name. For example, the keysym statement in a kmf-file (with **onF11** referencing to macro-file **onF11.mac**) may appear as

   KEY87 = MAC_onF11, 0xffc8; (XK_F11,XK_F11)

The Run Menu

This menu lets you launch a number of XLitePro's programs listed below. To run one of these programs, click its name on the **Run** menu of XServer or from the **X-clients** Program Folder of the package (when XServer is running). (In these cases, the programs are called with parameters specified by XServer.)

XServer Settings

The XSettings utility changes information stored in the ini-file, including settings used by XServer.

Note that you have to restart XServer after having changed any settings made by XSettings (in the ini-file).

Font Manager

The Font Manager allows you to view fonts accessible for X clients in the current X-session and change font settings.

Telnet

The Telnet utility starts terminal emulation with a remote system running the Telnet service.

XStartup

The Startup utility uses the REXEC, RSH, RLOGIN, or Telnet protocols to access hosts and then execute commands (e.g., to launch X clients).

ComSetup

The ComSetup utility makes communication settings relating to the TCP/IP transport interface.

Local X Clients

The XLitePro package includes the following local X and Motif clients:

- **mwm** (based on OpenMotif 2.0)
- **glxinfo** (the GLX extension local X Client)
- **glxgears** (the GLX extension local X Client)
- **xclock**
- **xev**
- **xfd**
- **xfontsel**
- **xkill**
- **xlsfonts**
- **xmodmap**
- **xprop**
- **xrdb**
- **xset**
- **xshowcmap**
- **xwininfo**

They have the same functionality and command line syntax as related remote (UNIX) ones, but were developed to be run under MS Windows, so you can use them the same way.

To run one of these programs, click its name on the **Run** menu of XServer or from the **X-clients** Program Folder of the package (when XServer is running). (In these cases, the clients are called with parameters specified by XServer.)

You can take advantage of the local X clients when remote (UNIX) ones are inaccessible or inconvenient to run.

For more information about these programs, please refer to Appendix F **Local X and Motif Clients.**

Other Commands on the X Client's Menu

The Refresh Item

This option is only available in the **Multiple** window mode. This option causes all X clients to re-display their windows.

The Local Refresh Item

This option lets you re-display all X clients' windows locally (by XServer, not X clients).

The Restart XServer Item

This option lets you restart XServer with closing all X clients and re-reading the ini-file used for the current X-session (that stores all XLitePro's settings). The default ini-file is **xwp.ini**.

The Messages Item

This item displays server related information and error messages.

The About Item

This item displays information about your XLitePro package.

The Help Item

This item displays the on-line help text of this chapter.

The Local Window Manager Item

This run-time setting is used in the **Multiple** window mode to switch on the local Window manager (for example, after XDM/CDE starts with faults).

The Readme & Ordering Info Item

This item uses Notepad to display the **readme** file.

Running XServer with Command Line Parameters

The 'xini' parameter

You can launch XServer (and begin your X-session) with the 'xini' command line parameter:

PATH\xserver.exe -xini <IniFilePath>

where <IniFilePath> specifies a full path to a specific ini-file and PATH indicates your XLitePro home directory.

This feature allows you to run several X-sessions each with its own ini-file (i.e. settings).

In order to do so, you can create a new X-session shortcut (e.g., in the XLitePro Programs' folder) and fill in the **Target** field in Properties of it with the command. For example, the **Target** field in Properties of the X-session shortcut might contain the following line:

"PATH\xserver.exe" -xini myxwp1.ini

By default, the field contains a call of XServer with no arguments, and the **xwp.ini** file will be used in this case.

To create your specific ini-file, you can copy the **xwp.ini** file and then change required parameters with the XSettings utility by starting it with the same command line parameter (see section **Running XSettings with Command Line Parameters** in Chapter **Configuring XLitePro**).

Note: When launched from the XServer/Run menu, ComSetup, XSettings, Startup, and Telnet use the same ini-file as XServer does.

Running Several X-sessions

Different X-sessions running simultaneously on your PC must use different Display Numbers. So you have to launch each X-session with a proper ini-file specified in a command line when invoking XServer. These ini-files must differ at least by Display Numbers. You may specify other settings of the ini-files as different as you need for particular X-sessions.

Example

1. Create two copies of your current ini-file, **xwp.ini**, with names **session0.ini** and **session1.ini**
2. Create two copies of the XSettings shortcut with names XSettings0 and XSettings1 and specify the ini-files for them respectively
3. Launch XSettings with XSettings0, specify 0 for Display Number, and exit
4. Launch XSettings with XSettings1, specify 1 for Display Number, and exit
5. Create two copies of the XSession shortcut with names XSession0 and XSession1 and specify the ini-files for them respectively.

Another way (without creating two XSettings shortcuts) is as follows:

1. Create two copies of your current ini-file, **xwp.ini**, with names **session0.ini** and **session1.ini**
2. Create two copies of the XSession shortcut with names XSession0 and XSession1 and specify the ini-files for them respectively
3. Launch X-session with XSession0, launch XSettings from the XServer/Run menu, specify 0 for Display Number, and exit from both programs
4. Launch X-session with XSession1, launch XSettings from the XServer/Run menu, specify 1 for Display Number, and exit from both programs

Now you can run these two XSessions at the same time - XSession0 with Display Number 0 and XSession1 with Display Number 1 (both for your display).

If you want three XSessions to run simultaneously, then, in addition to above:

1. Create a copy of one of your ini-files and give it a name **session2.ini**
2. Create a copy of the XSettings shortcut with a name XSettings2 and specify the ini-file for it
3. Launch XSettings with XSettings2, specify 2 for Display Number, and exit
4. Create a copy of the XSession shortcut with a name XSession2 and specify the ini-files for it.

Then you may run these three XSessions at the same time - XSession0 with Display Number 0, XSession1 with Display Number 1 and XSession2 with Display Number 2 (all for your display).

To specify an ini-file in these shortcuts, see section **Running XServer with Command Line Parameters** in this chapter.

Another way to launch several X-sessions simultaneously is to use the **DynamicDisplayNumber**, **XTcpPort**, and **XsessionTitle** settings described in section **The "[XSETUP]" Section of the ini-file** below.

Appendix B Description of Terminal Capabilities

Description of Terminal Capabilities

terminfo.ini is the ASCII file that describes the emulation capabilities of terminals. The description is very similar to the TERMINFO source code of UNIX system.

The file consists of terminal sections. Each section begins with a header string - a logical terminal name enclosed in brackets. The name is used to select the terminal type in the Telnet Options/ Settings/Type tab.

The section header is followed by a set of entry records that describes the emulation capabilities of the terminal. Each record consists of a capability keyword, a '=' separator, and one or more capability values separated by a ',' delimiter. White spaces are ignored after the ',' separator as well as after the last value.

Any line may contain a comment. A comment begins with a ';' character on a line and lasts to the end of the current line.

Normally there may be several groups of records in each terminal section:

- A group of terminal capability definitions (e.g., Lines, Columns, Colors, UseCRT, MG0, MG1, RGB, UseCSI, Use2W, Use2HW, Tab8, TabSet).
- A group of X-keys definitions (for non-standard keys of keyboards, e.g., auxiliary keypads or function keys). An X-keysym name followed by a '=' separator begins the record. X-keys definitions may have a single value for the PC-layout mode, or a single value for the VT-layout mode, or both. A '/' sign must precede the VT-layout mode value. In case of both modes, the first value is for the PC-layout mode. A value is a code sequence of one or more key codes (separated by a ',' delimiter) to be transmitted when you use the X-key defined. A code is a decimal number (to represent valid escape or control sequences for the terminal).

  Examples for VT240:

  XK_Home=155,72 / 155,50,126 ; CSI H / CSI 2 ~

  XK_KP_Home=155,72/143,119 ; CSI H / SS3 w

  XK_KP_Space=/143,117 ; / SS3 u

- A group of user defined keys (UDK) definitions. You can reprogram them in the Telnet Options/ Settings/User Defined tab. A UDK's name followed by a '=' separator begins the record. A value may be a sequence of one or more characters and/or hex codes (without any separator) to be transmitted when you use the UDK defined.

Examples:

Alt-F10=PRINT ; five characters

Shift-F1=0x1bOp ; one hex code 0x1b with two characters O and p

Terminal Parameter Settings

You can set up terminal capabilities by records with the following keywords and values:

Lines=number

The record specifies a number of lines on a screen of a terminal.

(Example: Lines=24)

Columns=number

The record specifies a number of characters in a line.

(Example: Columns=80)

Colors=number

The record specifies the maximum number of colors on the screen.

(Example: Colors=16)

RGB=R.G.B, ...

The record specifies the palette description for colors to be used for color terminals instead of default colors of MS Windows. Each color number (beginning at 0 to Colors-1) is represented by R, G and B - the relative intensities for red, green, and blue primaries (each in the range of 0...255 decimal) to be used for a certain color. The RGB values indicate normal locations in color space. The primaries in a triple are '.' separated while triples are separated by a ',' delimiter.

UseCSI=1

The terminal can (UseCSI=1) or cannot (UseCSI=0) use 8-bit control sequences.

UseCRT=1

The record defines to use the terminal or application mode for arrows keys instead of ANSI mode (as default).

Use2W=1

The terminal can (Use2W=1) or cannot (Use2W=0) use characters with double width. (See the control sequence Esc#6 for DEC terminals.)

Use2HW=1

The terminal can (Use2HW=1) or cannot (Use2HW=0) use characters with double height and width. (See control sequences Esc#3 and Esc#4 for DEC terminals.)

Tab8=1

The terminal can use the 8-space default tab stops (in case the TabSet record below does not exist).

TabSet=n1,n2,...

The record defines the horizontal tabulation stop set that will be used as the default tabulation set for the terminal. Decimal numbers separated by a ',' delimiter (n1,n2,...) are column numbers for horizontal tab stops. No values means the tabulation set is empty.

MG0i=inp.out,...
MG1i=inp.out,...

The records define re-mapping tables for the main character set (MG0) and for the alternate character set (MG1). Each table consists of code pairs separated by a ',' delimiter. A code pair has an input code (to be remapped), a '.' sign, and an output code from the character set used to display characters. The codes must be a hex number. Tables may continue onto multiple lines. The 'i' suffix in the keywords (MG0i or MG1i) shows the value for 'line number - 1'.

- Example for AT386 (9 pairs for MG0 on 2 lines):

  MG00=0x90.0xC9,0x91.0xBB,0x92.0xBC,0x93.0xC8,0x94.0xCD;

  MG01=0x95.0xBA,0x97.0xB9,0x98.0xCA,0x99.0xCC;

- Example for DEC terminals and XTERM (15 pairs for MG1 on 3 lines):

  MG10=0x6A.0xD9,0x6B.0xBF,0x6C.0xDA,0x6D.0xC0,0x6E.0xC5;

  MG11=0x6F.0xC4,0x70.0xC4,0x71.0xC4,0x72.0xC4,0x73.0xC4;

  MG12=0x74.0xC3,0x75.0xB4,0x76.0xC1,0x77.0xC2,0x78.0xB3;

< previous | content | next >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

Appendix C Troubleshooting

Here are the answers to some frequently asked and particular questions.

XServer & XSettings & Font Service

I am able to get xterm on my Unix box, but I am unable to get xterm on my PC when I telnet-ssh to my Linux box running Kubuntu. When I try to run xterm from the telnet-ssh console on my PC I get the "Error of failed request" message.

The reason of this error message is absence of the font requested by xterm on your Kubuntu machine:
"-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso10646-1".
One simple solution is to connect your XServer to any font server that supports these fonts (see section **Font Server** in chapter **Font Control**). Another solution is to download the Unicode ISO10646 fontset from Internet sites and include it in Font Path for your XServer to use it (see section **Font Directory** in chapter **Font Control**).

If I start one Xsession as normal privileged user and then afterwards another Xsession logged in as administrator they will both get DisplayNumber 0 which is pretty bad.

Yes, this is true. This may take place under Win2003 (and under WinXP with especial settings).

The **Auto-DisplayNumber** mode requires to create global Windows named memory objects to store necessary information. The default Vista security state denies to create such a memory object. Also, non-privileged users are not allowed to create shareable Windows objects.

You have two ways to avoid this:
1) Start XServer firstly from the administrative user (as you did)
2) Allow INTERACTIVE users to **Create global objects**.
To do this, start the **Administrative tools - Local Security Settings - User Rights Assignment** dialog, choose **Property** of the **Create global objects** policy and add the INTERACTIVE group to the group list (it may already contain the ADMINISTRATORS and SERVICE groups).

Is it possible to convert fonts used in Motif to be used with XLitePro (the fonts with the .pcf suffix)?

Yes, it is. You should create a directory for **.pcf** files, create the **fonts.dir** and **fonts.ali** files in it, and add the corresponding item to Font Path (the **Priority ordered path** box) by using XSettings. XServer makes use of **.pcf** fonts the same way as **.snf** fonts.

You can enable **Trace Fonts Requests** in XSettings to find out fonts that are absent (from the **xserver.out** file).

There are several ways to add fonts to XLitePro (see Chapter **Font Control**):

1) Connect to a remote font server which provides fonts you need.

You can launch XSettings, press the **FontPath** button, enter your font server specification in the **Font directory** field, select a font server specification in the **Priority ordered path** box, and **Insert** the Font directory into the path list before/after the selected item.

To detect required font directories and their order on a remote host, please logon to the remote host by using the console (under the same account), and then run the command line "**xset -q**" from the xterm window or from the console. Please store the "Font Path:" from the output of this command.

You should add the missing paths to these fonts in the same order into the "catalogue=" line of the font server's config-file. Then, if the font server is running, restart it or re-read its config-file.

E.g., the config-file for Solaris is **/usr/openwin/lib/X11/fontserver.cfg**, for Linux **/usr/X11R6/lib/X11/fs/config**.

Note: If you are using a font server located on a Linux host, you cannot check its config-file because all fonts are included in the config-file.

If you are using X11 Forwarding for X-applications, you should use the forwarding for port 7100 of a font server too.

2) Copy fonts you need (**.snf** or **.pcf** files only) into a directory on your PC and include its name into the Font Path list.

Example for Asian fonts:

If you have Chinese, Japanese, Korean, or other 16-bit fonts supported by a UNIX host, you should:

1. Uncompress required font files (**.pcf** or **.snf** files only) to a new directory on your host (say, Asianfonts)
2. Run the "**mkfontdir**" utility for this directory to create the **fonts.dir** file.

   To map new names to existing fonts, you should create the **fonts.alias** file (or edit the existing one).

3. Create a new directory under the FONTS folder on your PC (say, .../FONTS/Asianfnt) to contain new fonts
4. Copy all binary font files (**.pcf** or **.snf** files only) from the Asianfonts directory of your host to the .../FONTS/Asianfnt directory on your PC (e.g., by ftp in the binary mode)
5. Copy the **fonts.dir** text file from the Asianfonts directory of your host to the .../FONTS/Asianfnt directory on your PC (e.g., by ftp in the ASCII mode)
6. Copy the **fonts.alias** text file from the Asianfonts directory of your host to the .../FONTS/Asianfnt directory on your PC (e.g., by ftp in the ASCII mode) as the **fonts.ali** file
7. Add the path to the .../FONTS/Asianfnt directory on your PC to Font Path.

   You can launch XSettings, press the **Font Path** button, enter the directory name in the **Font directory** field or select it by using **Browse**, select a font source specification in the **Priority ordered path** box, and **Insert** the Font directory into the path list **before/after** the selected item.

8. Check the .../FONTS/Asianfnt directory by using the XFontset Service utility.

   Then, you can view the **Entire Fonts List** of the .../FONTS/Asianfnt directory.

3) Use XSettings' PSEUDO fonts feature to emulate X-fonts with MS Windows fonts which actually exist on your PC.

4) Create aliases for fonts you need in a directory to substitute them with MS Windows fonts which actually exist on your PC and include the directory name into the Font Path list.

For example, the 'sun-fontspecific' fonts may be substituted with the 'adobe-fontspecific' fonts.

5) Edit the **fonts.ali** file in any existing font directory and create aliases for absent fonts to substitute them with existing fonts.

Of course, the first and second ways are preferable.

You can try to run the following command line

fsinfo -server fontserverhost:portnumber

If you get the message:

'/usr/openwin/bin/fsinfo: unable to open server "fontserverhost:portnumber'

then check:

1. that the following font server line is present in the **/etc/services** file:

   fs 7100/tcp # Font server

2. that this port (7100 in our case) is being listened on;

and run the following command on your host:

netstat -an | grep 7100

If you do not get any answer, then check configuration of the X font server on your host.

When trying to start some of our local UNIX applications, the following error was displayed - Cannot accept '/usr/local/Apstools/12.1/lib/fonts' FontPath Element.

You need to avoid this problem. While executing, your application adds a new item '/usr/local/Apstools/12.1/lib/fonts' to XServer's Font Path. This action is not legal for XLitePro's XServer because it works only with local Font Path and/or with a remote font server path.

When using the X Display manager the font that is on the screen is the same as the one I am using now. But when I log in from the host or HP Xterminal the font looks like a 'sans serif' or 'arial' type of font. How can I get the required font? I noted down the fonts that are loaded

by an Hp-Xterminal when it boots up from the host. They are /usr/lib/X11/700X/fonts/hp-roman8/75dpi/system19.snf and system23.snf, system16.snf, system17.snf, system13.snf, ser11x19.snf. I copied these fonts to my local PC and added these fonts to the font path, but when I started the X Display manager, I received garbage on the screen, where the login and password words were before.

The HP-snf font format actually differs from the X11-snf font format. XLitePro's X-session processes only X11-snf fonts and cannot process HP-snf fonts. X-session cannot distinguish between these formats because they have the same file name extension.

You have to convert the HP-snf to X11-snf.

Cut and Paste did not work between windows (e.g., between W'95 Notepad and SUN/UNIX Textedit). I have still to try dragging and dropping a file between windows (e.g., UNIX Filemanager to W'95 File system).

In your case, you should choose the **CLIPBOARD** X Selection to provide the transfer needed.

To use the cut&paste functions, you should know the global cut buffers (X Selections) that your X-application uses for these operations (from X-application manuals).

Different X-applications can use different X Selections for cut&paste. For example, normally,

- "xterm" uses the **PRIMARY** X Selection
- "cmdtool" uses the **CLIPBOARD** X Selection
- "dtterm" uses both (**PRIMARY** for mouse Button1/Button2 actions, **CLIPBOARD** for Edit menu's Copy/Paste actions).

Before cut&paste, you should correctly define the global cut buffer (X Selection) that XServer and your X-application will use.

For example, to copy text between xterm and MS WordPad, you should specify the **PRIMARY** X Selection for xterm and XServer before copying.

For the xterm window, you can define the buffer by choosing: Select Options/XSelection... **PRIMARY**/OK.

Make sure that the **Auto Clipboard Copy&Paste** check box in the XSettings dialog box is enabled. (See section **A Note on Copy and Paste** in chapter **Using XServer**.)

If you want to copy text from xterm to MS WordPad, you can do the following:

In the xterm window:

1. Move the mouse pointer to the beginning of the text, hold down the mouse Button1 (usually left) while moving the pointer to the end of the text, and then release the button. The selected text is highlighted and saved in the **PRIMARY** global cut buffer.

In the WordPad window:

2. Move the mouse pointer to the place where you want to insert the text from the buffer and

click the mouse Button1 (usually left).

3. Click the Edit menu, and then choose Paste (or Ctrl-V) to paste the text from the **CLIPBOARD** of MS Windows. The text selected in the xterm window will appear in the WordPad window.

If you want to copy text from MS WordPad to xterm, you can do the following:

In the WordPad window:

1. Move the mouse pointer to the beginning of the text, hold down the mouse Button1 (usually left) while moving the pointer to the end of the text, and then release the button. The selected text is highlighted.
2. Click the Edit menu, and then choose Copy (or Ctrl-C). The selected text is saved in the **CLIPBOARD** of MS Windows.

In the xterm window:

3. Move the mouse pointer to the place where you want to insert the text from the buffer and click the mouse Button2 to paste the text from the **PRIMARY** X Selection. The text selected in the WordPad window will appear in the xterm window.

The mouse Button2 is normally the middle button or simultaneously pressed left and right buttons if you have enabled the **Middle Button Emulation** check box in the XSettings dialog box.

I would like to know what True color support means... Will True color work with more than 256 colors?

True color support (in Windows) means that a video-driver supports the color palette with 64K colors. XServer supports 256-color palette and CAN operate with any High/True-color video-driver of Windows.

Is there a difference between using the X-manager and the windows manager in cutting and pasting?

XLitePro cannot write the X Selection/bitmap to a file.

Is it possible to print X application's windows to my local computer (MS Windows 98 printer) with XLitePro?

Yes, it is. On your local printer, you can print from MS Windows applications and from remote X applications as well (if the printer is accessible for them).

If you want to print an X application's window that you see on your display to a printer connected to your PC,

- Make sure that the X application's window is active on your display
- Press the **Alt+Print Screen** shortcut key to copy the window to MS Clipboard
- Paste MS Clipboard to a MS Windows application you want to use (e.g., MS Paint)

- Print the image from this application to your printer.

By using X Selections, you can transfer required text from your X application to a MS Windows application (e.g., from xterm to WordPad) and then print the whole document to your printer.

I want to run Netscape from my PC on the remote UNIX machine. I use 'your' telnet to log into the remote UNIX computer. There I can find my PC's IP address. I try to start 'xterm' on the UNIX machine. I think it does start in the 'background'. When I try to run, say, Netscape, 'it' complains that it cannot find 'display'. What the problem could be?

Make sure that your remote host is accessible from your PC. First, try to open a telnet connection to a remote host. If you can do that, then check whether you can start X-applications from the telnet session on your remote host:

- Start X-session on your PC
- Open the telnet connection to the remote host
- Launch any X-application from the telnet session:

e.g.,

/usr/bin/X11/xterm -display IP_address_of_your_PC:0&

or

/usr/bin/X11/xterm -display Name_of_your_PC:0&

(in this case, you should add the name and IP-address of your PC into the **/etc/hosts** file on your Unix host).

Has the xterm window appeared on the screen of your PC?

If not, check that TCP ports 6000 - 60xx used by XServer on your PC are accessible from the remote host.

If yes, you can try to launch X-applications by using the Startup utility. First:

- Find the configuration file of inetd, **/etc/inetd.conf**, (or xinetd) on the remote host (please find out which Internet service daemon is running on your remote host: inetd or xinetd)
- Check that the following lines in the **/etc/inetd.conf** file of the Unix host are present and are not commented-out (by default, these lines are commented-out):

  "exec ..." for REXEC (512/tcp) (if you want to use rexec in Startup)

  "shell ..." for RSH (514/tcp) (if you want to use rsh in Startup)

  "login...." for RLOGIN (513/tcp) (if you want to use rlogin in Startup)

If the xinetd daemon is used on your remote host, it is recommended that you read corresponding

manual pages for xinetd.

If these lines are present and are not commented-out, you can try to launch X-applications (or run any Unix command) by using Startup.

Launch Startup, select one of the start methods (REXEC/RSH/RLOGIN) supported by the Unix host (see the **/etc/inetd.conf** file of the Unix host), enter the Host Name or IP-address of the Unix host, User ID, and Password in corresponding fields of the Startup dialog box, enter any Unix command in the "Command" field (e.g. '**ls -l**'), and then click the **Execute** button.

Is there any output of this command in the "Status Messages" field?

If yes, then enter the command launching xterm (e.g., **/usr/X11R6/bin/xterm -display $(LOCALIP):0**) and click the **Execute** button.

Has the xterm window appeared on the screen of your PC?

Also, you can try to use the XDMCP mode of XServer. If you want to use XServer in the XDMCP mode, please check that the xdm-daemon is running on the remote host and check XDMCP configuration.

To use XDMCP, you should launch XServer in the **Single/Full** or **Multiple+RemoteWM** mode because, if XServer is running in the **Multiple** mode, some remote window managers (including 'gnome' and KDE) close X-session when they detect any other running window manager (e.g., MS Windows window manager or local 'mwm').

To configure XServer, you should start the XSettings utility, select the **Single/Full** or **Multiple +RemoteWM** mode, enable the **Use XDMCP** check box, click the **Settings** button, and then specify XDMCP settings.

Before using XDMCP, make sure that XDMCP on the remote host listens to UDP port 177. To do so, run the "**netstat -an|grep 177**" command on the remote host. You may see from its output:

"udp 0 0 0.0.0.0:177 0.0.0.0:* "

(or for Solaris station: *.177 Idle)

If you do not see this, please check configuration of XDMCP on your remote host.

Also, make sure that UDP port 177 on the remote host is accessible from your PC (i.e. there are no any firewalls, etc. filtering these packages between your PC and the remote host).

If UDP port 177 on the remote host is not accessible from your PC, you cannot use XDMCP mode of XServer.

NOTES:

When XServer is running in the **Single/Full** or **Multiple+RemoteWM** mode, the remote window manager that you select will be used (e.g., 'olwm', CDE, 'twm', 'mwm', etc.).

If you want to use a remote window manager, please use the **Single/Full** or **Multiple+RemoteWM** mode. In this case, you should use the XDMCP mode or start any remote window manager from the Unix host (e.g., start 'xterm' and then start the required window manager from the xterm window).

In this case, the remote window manager only provides the windows managing functions (moving, resizing, etc.).

Also, you can launch the local 'mwm' program (Motif Window Manager) on your PC from the X-session's Run menu (for the **Single/Full** or **Multiple+RemoteWM** mode). In this case, 'mwm' will provide the windows managing functions.

If you do not want to use a remote window manager, please use the **Multiple** mode. When XServer is running in the **Multiple** window mode, the MS Windows window manager is used and any other window manager is blocked.

Some remote window managers (including 'gnome' and KDE) close the X session if they detect any other running window manager. In this case, you cannot use the **Multiple** mode and should only use the **Single/Full** or **Multiple+RemoteWM** window mode.

17. Font Control **< previous | content | next >**

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

Appendix F

Local X and Motif Clients

The XLitePro package includes the following local X and Motif clients:

- **mwm** (based on OpenMotif 2.0)
- **glxinfo** (the GLX extension local X Client)
- **glxgears** (the GLX extension local X Client)
- **xclock**
- **xdpyinfo**
- **xev**
- **xfd**
- **xfontsel**
- **xkill**
- **xlsfonts**
- **xmodmap**
- **xprop**
- **xrdb**
- **xset**
- **xshowcmap**
- **xwininfo**

They have the same functionality and command line syntax as related remote (UNIX) ones, but were developed to be run under MS Windows, so you can use them the same way.

To run one of these programs, click its name on the **X-clients** Program Folder of the package (when XServer is running) or from the **Run** menu of XServer. (In these cases, the clients are called with parameters specified by XServer.)

Also, you can launch the clients from Startup (in the **lShell** mode) or from the **Start/Run** menu of your MS Windows system. (In this case, you must specify startup parameters for the invoked X client on the command line.)

You can take advantage of the local X clients when remote (UNIX) ones are inaccessible or inconvenient to run.

The following is brief descriptions of available local X clients. For more information, refer to corresponding manuals on these applications.

mwm

mwm is a local implementation of the Motif window manager (based on OpenMotif 2.0). The local **mwm** program may be useful when XServer is in the **Single Window** mode (this allows not to use a remote window manager).

glxinfo

glxinfo is a sample program for displaying info about a GLX extension and OpenGL renderer. It lists information about the GLX extension, OpenGL capable visuals, and the OpenGL renderer on an X server. The GLX and renderer info includes the version and extension attributes. The visual info lists the GLX visual attributes available for each OpenGL capable visual (e.g. whether the visual is double buffered, the component sizes, Z-buffering depth, etc). By default the visual info is presented in a concise 80 character wide tabular format. The **-l** option allows you to print interesting OpenGL limits.

glxgears

The **glxgears** program is a sample GLX version of the "gears" GL demo. It is a GLX demo that draws three rotating gears, and prints out framerate information to stdout. The **-info** option allows you to print out GL implementation information before running the demo.

xclock

The **xclock** program displays the time in analog or digital form. The time is continuously updated at a frequency that may be specified by the user.

xdpyinfo

The **xdpyinfo** program displays information about an X server. It is used to examine the capabilities of a server, the predefined values for various parameters used in communicating between clients and the server, and the different types of screens and visuals that are available.

xev

xev is the X Event Tester. You can run this standard X client locally (on your PC) to obtain the values of keysyms for keys. According to its manual, when **xev** is running, it creates a window and then asks the X server to send it notices called 'events' whenever anything happens to the window (such as being moved, resized, typed in, clicked in, mouse movements, button clicks, etc.). It is useful for seeing what causes events to occur and to display the information that they contain. For example, you can run **xev** to obtain the values of keysyms for key presses.

xfd

This program is used to displays all the characters in an X font. According to its manual, the characters are displayed in a grid of boxes, each large enough to hold any single character in the font. Individual character metrics (index, width, bearings, ascent and descent) can be displayed at the top of the window by clicking on the desired character.

xfontsel

The **xfontsel** application provides a simple way to display the fonts known to your X server, examine samples of each, and retrieve the X Logical Font Description ("XLFD") full name for a font (i.e., **xfontsel** is a point & click interface for selecting X11 font names).

According to its manual, if the **pattern** option is not specified, all fonts with XLFD 14-part names will be selectable. To work with only a subset of the fonts, you can specify **-pattern** followed by a

partially or fully qualified font name (e.g., "**-pattern *medium***" will select that subset of fonts which contain the string "medium" somewhere in their font name). Be careful about escaping wildcard characters in your shell.

The **sample** option specifies the sample text to be used to display the selected font if the font is linearly indexed, overriding the default. The **sample16** option specifies the sample text to be used to display the selected font if the font is matrix encoded, overriding the default. The **noscaled** option disables the ability to select scaled fonts at arbitrary pixel or point sizes. This makes it clear which bitmap sizes are advertised by the server, and can avoid an accidental and sometimes prolonged wait for a font to be scaled.

Clicking any pointer button in one of the XLFD field names will pop up a menu of the currently known possibilities for that field. If previous choices of other fields were made, only values for fonts that matched the previously selected fields will be selectable. To make other values selectable, you must deselect some other field(s) by choosing the '*' entry in that field.

Scalable fonts come back from the server with zero for the pixel size, point size, and average width fields. Selecting a font name with a zero in these positions results in an implementation-dependent size. Any pixel or point size can be selected to scale the font to a particular size. Any average width can be selected to anamorphically scale the font (although you may find this challenging given the size of the average width menu).

Clicking the left pointer button in the select widget will cause the currently selected font name to become the PRIMARY text selection as well as the PRIMARY_FONT selection. This then allows you to paste the string into other applications. The select button remains highlighted to remind you of this fact, and de-highlights when some other application takes the PRIMARY selection away.

xkill

The **xkill** program is a utility for forcing the X server to close connections to clients. This program is very dangerous, but is useful for aborting programs that have displayed undesired windows on a user's screen. According to its manual, if no resource identifier is given with -id, **xkill** will display a special cursor as a prompt for the user to select a window to be killed. If a pointer button is pressed over a non-root window, the server will close its connection to the client that created the window.

xlsfonts

This program is used to display server font list for X. According to its manual, **xlsfonts** lists the fonts that match the given pattern. The wildcard character '*' may be used to match any sequence of characters (including none), and '?' to match any single character. If no pattern is given, '*' is assumed. The '*' and '?' characters must be quoted to prevent them from being expanded by the shell.

The **l** option lists some attributes of the font on one line in addition to its name. The **ll** option lists font properties in addition to **l** output. The **lll** option lists character metrics in addition to **ll** output.

xmodmap

The **xmodmap** utility is for modifying keymaps and pointer button mappings in X. It is used to edit and display the keyboard modifier map and keymap table that are used by client applications to convert event keycodes into keysyms. It is usually run from the user's session startup script to configure the keyboard according to personal tastes.

Some options that may be used with xmodmap (according to the manual):

- The **-display display** option specifies the host and display to use.
- The **-e expression** option specifies an expression to be executed. Any number of expressions may be specified from the command line.
- The **-pm** option indicates that the current modifier map should be printed on the standard output.
- The **-pk** option indicates that the current keymap table should be printed on the standard output.
- The **-pke** option indicates that the current keymap table should be printed on the standard output in the form of expressions that can be fed back to **xmodmap**.
- The **-pp** option indicates that the current pointer map should be printed on the standard output.

The **filename** argument specifies a file containing xmodmap expressions to be executed. This file is usually kept in the user's home directory with a name like **.xmodmaprc**.

The **xmodmap** program reads a list of expressions (from a command line or a file) and parses them all before attempting to execute any of them. This makes it possible to refer to keysyms that are being redefined in a natural way without having to worry as much about name conflicts.

The list of keysym names may be found in the header file **X11/keysymdef.h** (without the XK_ prefix) or the keysym database **<XRoot>/lib/X11/XKeysymDB**, where <XRoot> refers to the root of the X11 install tree.

Example:
Many pointers are designed such that the first button is pressed using the index finger of the right hand. People who are left-handed frequently find that it is more comfortable to reverse the button codes that get generated so that the primary button is pressed using the index finger of the left hand. This could be done on a 3-button pointer as follows:

```
%  xmodmap -e "pointer = 3 2 1"
```

xprop

The **xprop** utility is for displaying window and font properties in an X server. According to its manual, one window or font is selected using the command line arguments or possibly in the case of an X's window, by clicking on the desired window. A list of properties is then given, possibly with formatting information. Normally each property name is displayed by printing first the property name then its type (if it has one) in parentheses followed by its value.

xrdb

The **xrdb** utility is for controlling the X server resource database. According to its manual, **xrdb** is used to get or set the contents of the RESOURCE_MANAGER property on the root window of screen 0, or the SCREEN_RESOURCES property on the root window of any or all screens, or everything combined. You would normally run this program from your X startup file.

Most X clients use the screen-independent resource property, RESOURCE_MANAGER, and the screen-specific resource property, SCREEN_RESOURCES, to get user preferences about color, fonts, and so on for applications. The RESOURCE_MANAGER property is used for resources that apply to all screens of the display. The SCREEN_RESOURCES property on each screen specifies additional (or overriding) resources to be used for that screen. When there is only one screen, SCREEN_RESOURCES is normally not used and all resources are just placed in the RESOURCE_MANAGER property.

xset

This program is used to set various user preference options of the display. According to its manual, the options control the bell parameters, key click, autorepeat, the turning on/off the keyboard LEDs, the mouse parameters, pixel color values, the screen saver parameters, the font path, Energy Star mode, power management, and many others.

xshowcmap

The **xshowcmap** program displays the contents of the currently active colormap in a window (a map of the system colors). The created window shows a square for every color currently defined in the server's active colormap. The number of squares is the number of colormap-cells the server supports.

xwininfo

The **xwininfo** utility is for displaying information about X's windows. According to its manual, various information is displayed depending on which options are selected. The user has the option of selecting the target window with the mouse or by specifying its window id or name. There are also special options to quickly obtain information on the screen's root window or on parent and/or children windows.

The **stats** (default) option causes the display of various attributes pertaining to the location and appearance of the selected window. Information displayed includes the location of the window, its width and height, its depth, border width, class, colormap id if any, map state, backing-store hint, and location of the corners.

The **bits** option causes the display of various attributes pertaining to the selected window's raw bits and how the selected window is to be stored. Displayed information includes the selected window's bit gravity, window gravity, backing-store hint, backing-planes value, backing pixel, and whether or not the window has save-under set.

The **metric** option causes all individual height, width, and x and y positions to be displayed in millimetres as well as number of pixels, based on what the server thinks the resolution is.

< previous | content

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us