

PortShare Software Quick Start Guide

Use with LES1101A, LES1102A, LES1108A, LES1116A, LES1132A, LES1148A, LES1208A, LES1216A, LES1232A, LES1248A, LES1202A, LES1203A-11G, LES1204A-3G, LES1203A-M, and LES1204A Secure Device Servers and Console Servers, and Remote Console Managers.

PortShare Quick Start Guide

TRADEMARKS USED IN THIS MANUAL

Black Box and the Double Diamond logo are registered trademarks of BB Technologies, Inc.

AIX is a registered trademark of International Business Machines Corporation.

HP-UX is a registered trademark of Hewlett-Packard.

Linux is a registered trademark of Linus Torvalds.

Nagios is a registered trademark of Nagios Enterprises.

SCO is a registered trademark of The SCO Group, Inc.

Solaris is a trademark of Sun Microsystems, Inc.

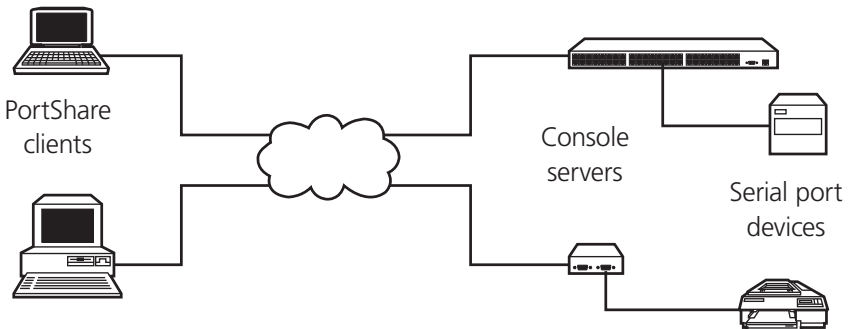
Unixware is a registered trademark of X/Open Company, Ltd.

Windows is a registered trademark of Microsoft Corporation.

Any other trademarks mentioned in this manual are acknowledged to be the property of the trademark owners.

About PortShare Software

Port Share software delivers the virtual serial port technology your Windows® and Linux® applications need to open remote serial ports and read the data from serial devices that are connected to your console server.

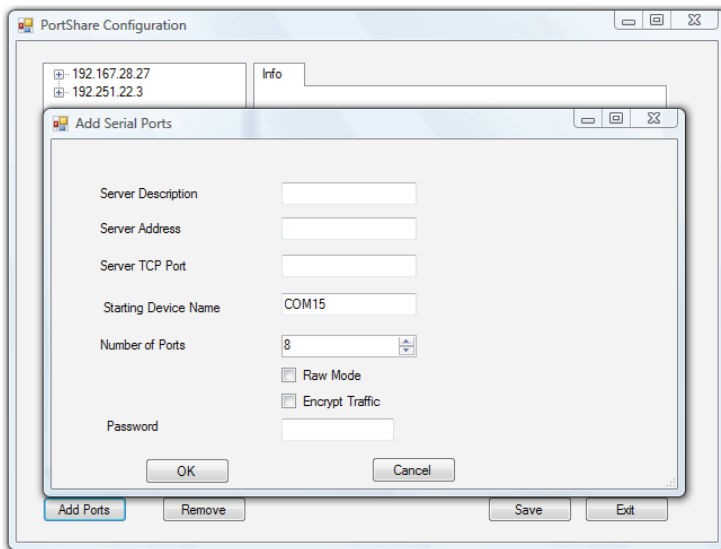


PortShare is supplied free with each console server, and you are licensed to install it on one or more computers for accessing any serial device connected to a console server port. This Quick Start walks you through basic configuration. For more details, refer to the PortShare chapter in the user's manual on the CD-ROM.

PortShare for Windows

The portshare_setup.exe program is included on the CD supplied with your console server (or you can download a copy for free from the ftp site).

1. Double-click to start installation.
2. Click the "PortShare" icon on your desktop to start the client.



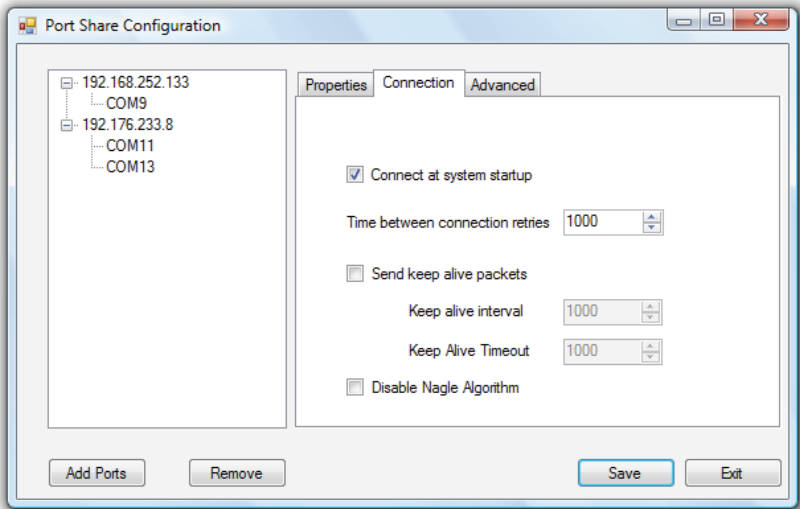
3. Click on "Add Ports" and specify a name to identify the connection in the "Server Description" tab.
 4. Enter the console server's IP address (or network name).
 5. Enter the Server TCP Port number that matches the port you have configured for the serial device on the remote console server. Make sure this port isn't blocked by a firewall.
- Telnet RFC2217 mode is configured by default, so the range of port numbers available on a 16-port console server would be 5001–5016.

PortShare Quick Start Guide

- Alternately check RAW mode (4001–4048 on a 48-port console server).

NOTE: Encrypted mode enables SSL/TLS encryption of the data going to the port and you will need to enter a Password. This mode requires firmware 3.1 in the console server.

6. Select the starting COM port (COM1 to COM4096) and the number of ports to be added (sequential port numbers will be assigned automatically). Click "OK."
7. To configure a COM port, simply click on the desired COMx label.
8. You can then configure the COM port in the Connection and Advanced windows:

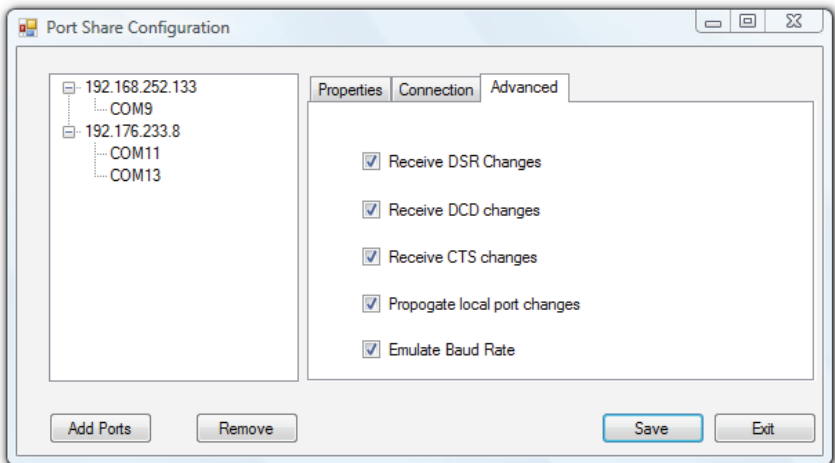


- Connect at system startup—When enabled, PortShare will try to connect to the console server as the PortShare service starts (instead of waiting for the application to open).
- Time Between Connection Retries specifies the number of seconds between TCP connection retries after a client-initiated connection failure. Valid values are 1–255. (The default is 1 second, and PortShare will continue attempting to reconnect indefinitely to the console server at this interval.)

- The Send Keep-Alive Packets option tests if the TCP connection is still up when no data has been sent for a while by sending keep-alive messages. Select this option and specify a period of time (in milliseconds) after which PortShare sends a command to the remote console server end to verify the connection's integrity and keep the connection alive.
- The Keep Alive Interval specifies the number of seconds to wait on an idle connection before sending a keep-alive message. The default is 1 second. The Keep Alive Timeout specifies how long PortShare should wait for a keep-alive response before timing out the connection.

NOTE: Make sure the remote serial device is connected to the nominated port on your remote console server and the serial port has been configured. That is, set the RS-232 Common Settings such as baud rate, select Console server mode for the port, and specify the appropriate protocol to be used (RAW TCP , RFC2217, or PortShare Secure mode for encrypted communication). Also make sure you can access the console server.

- Disable Nagle Algorithm—the Nagle Algorithm is enabled by default and it reduces the number of small packets sent by PortShare across the network.
- Check Receive DSR/DCD/CTS changes if the flow control signal status from the physical serial port on the console server is to be reflected back to the Windows COM port driver (some serial communications applications prefer to run without any hardware flow control, i.e. in “two-wire” mode).



- The Propagate Local Port Changes allows complete serial device control by the Windows application so it operates exactly like a directly connected serial COM port. It provides a complete COM port interface between the attached serial device and the network, providing hardware and software flow control. The baud rate etc of the remote serial port is controlled by the settings for that COM port on a Windows computer. If not selected, then the port serial configuration parameters are set on the console server.
- With the Emulate Baud Rate selected, PortShare will only send data out at the baud rate configured by the local Application using the COM port.

PortShare for Linux

The PortShare driver for Linux maps the console server serial port to a host TTY port.

The portshare-serial-client is an open source utility for Linux, AIX®, HP-UX®, SCO®, Solaris™ and UnixWare®. This utility can be freely downloaded from the ftp site.

This PortShare serial port redirector allows you to use a serial device connected to the remote console server as if it were connected to your local serial port. The portshare-serial-client creates a pseudo TTY port, connects the serial application to the pseudo TTY port, receives data from the pseudo TTY port, transmits it to the console server through the network, and receives data from the console server through the network and transmits it to the pseudo-TTY port.

You first need to set up the console server and attach and configure the remote serial port device. That is, make sure the console server IP configuration is OK and that you can access the unit (ping, telnet...) and configure the console server serial port to RAW or RFC2217 mode.

To install the PortShare serial client:

1. Build and install the package (as root):

```
$ ./configure && make && make install
```

Note that the '--prefix=' option is ignored by configure.

2. Configure the devices by editing /etc/portshare-devices (sample configurations and the format are shown next).
3. Start the portshare devices:

```
/usr/local/sbin/portshare-serial-client start
```

Examples showing the virtual port configuration and using a remote console serial port as a local TTY port on the Linux host.:

/etc/portshare-devices

devname:rastype:rasname:physport:type:options where :

devname -> Device full pathname

rastype -> Console server type (e.g. cm4008 or im4004 or acm5003)

rasname -> Console server host name or IP address

physport -> Physical port number on console server

type -> Server type : rfc2217 or socket (raw TCP)

opts -> per-port interface options (optional)

To connect via a secure SSH tunnel, use the '-P' parameter as part of "opts", and give the TCP port number used for the local end of the tunnel. For example, "-P 22222" will attempt to connect to local TCP port 22222. Also set the rasname to "localhost". The SSH tunnel must already be set up for this to work.

- Connect to Port 1 on a 48-port console server at IP address 10.111.254.1, using RFC2217:

/dev/otty01:cm4148:10.111.254.1:1:rfc2217

- Connect to Port 8 on a 8-port console server at IP address 10.111.254.2, using RFC2217:

/dev/otty02:cm4008:10.111.254.2:8:rfc2217

- Create an SSH tunnel from localhost to a console server. This tunnel connects to Serial Port 3 on the console server and uses rfc2217. The rfc2217 TCP port base on the console manager has been set to an alternate base of 9000. The local TCP port used for the tunnel is 12345:

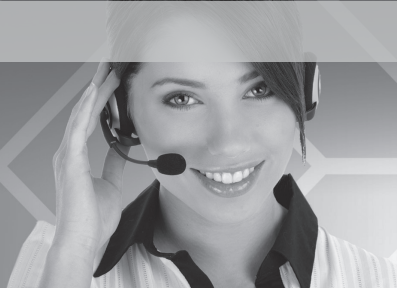
ssh -L 12345:10.111.254.3:9003 <username>@10.111.254.3 -N

Use this tunnel to make the connection:

/dev/otty03:cm4008:localhost:3:rfc2217:-P 12345

Black Box Tech Support: FREE! Live. 24/7.

Tech support the
way it should be.



Great tech support is just 20 seconds away at
724-746-5500 or blackbox.com.



About Black Box

Black Box Network Services is your source an extensive range of networking and infrastructure products. You'll find everything from cabinets and racks and power and surge protection products to media converters and Ethernet switches all supported by free, live 24/7 Tech support available in 20 seconds or less.

© Copyright 2011. All rights reserved.