USER MANUAL

LES1500, LES1600, LES1700-R2 SERIES

LES SERIES CONSOLE SERVERS

24/7 TECHNICAL SUPPORT AT 1.877.877.2269 OR VISIT BLACKBOX.COM

BLACK BOX®

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

REVISION HISTORY

RELEASE: V6.38

SAFETY PRECAUTIONS

Follow the safety precautions below when installing and operating the console server:

- Do not remove the metal covers. There are no operator-serviceable components inside. Opening or removing the cover may expose you to dangerous voltage that may cause fire or electric shock. Refer all service to Black Box-qualified personnel.

To avoid electric shock, the power cord protective grounding conductor must be connected through to ground.

- Always pull on the plug, not the cable, when disconnecting the power cord from the socket.
- Do not connect or disconnect the console server during an electrical storm.
- We recommend that you use a surge suppressor or UPS to protect the equipment from transients.
- Proper back-up systems and necessary safety devices should be used to protect against injury, death or property damage due to system failure. Such protection is the responsibility of the user.
- This console server device is not approved for use as a life-support or medical system.
- Any changes or modifications made to this console server device without the explicit approval and consent of Black Box will void Black Box of any liability or responsibility of injury or loss caused by any malfunction.
- This equipment is for indoor use only. All the console's communication wirings are limited to use inside of a building.

ABOUT THIS MANUAL

PRODUCTS COVERED

The Black Box User Manual describes the features and capabilities of the following Black Box product products, and provides instructions to best take advantage of them:

- LES1500 Series Console Servers: LES1516A, LES1532A, LES1548A
- LES1600 Series Console Servers: LES1604A, LES1604A-R-R2, LES1608A
- LES1700-R2 Series Console Servers: LES1708A-R2, LES1716A-R2, LES1732A-R2, LES1748A-R2

Each of these products is referred to generically in this manual as a console server.

Where appropriate, product groups may be referred to as console servers, gateways or by specific product line name or product group (for example the LES1500 family).

WHO SHOULD READ THIS USER MANUAL?

You should read this manual if you are responsible for evaluating, installing, operating, or managing a Black Box appliance. This manual assumes you are familiar with the internal network of your organization, and are familiar with the Internet, IP networks, HTTP, FTP and basic security operations.

MANUAL ORGANIZATION

The Black Box User Manual is structured as follows:

Safety Precautions

1. Specifications: Technical specifications for the console servers.

2. Overview: An overview of the console server's features and information regarding this manual.

3. Installation: Physical installation of the console server and the interconnecting of managed devices.

4. System configuration: Initial installation and configuration of the console server and the supported services.

5. Serial port, host, device and user configuration: Configuring serial ports and connected network hosts, and setting up users.

6. Firewall, failover, and OOB access: Set up the firewall and the high availability access features of the console server.

7. SSH tunnels and SDT connector: Secure remote access using SSH and configure for RDP, VNC, HTTP, HTTPS and access to network- and serially-connected devices.

8. Alerts, auto-response and logging: Set up local and remote event and data logs. Configure auto-responses to trigger events.

9. Power, environment and digital I/O: Manage USB, serial and network attached power strips and UPS supplies. Also EMD environmental sensor configuration.

10. Authentication: Access to the console server requires authenticated usernames and passwords.

11. Nagios Integration: Set Nagios central management. Configure console server as a distributed Nagios server.

12. System Management: Access to and configuration of services to be run on the console server.

13. Status reports: The dashboard summary and detailed status and logs of serial and network connected devices (ports, hosts, power and environment).

14. Management: Port controls and user-accessible reports.

15. Configuration from the command line: Command-line installation and configuration using the config command.

16. Advanced Configuration: Advanced command-line configuration activities using Linux commands.

17. Appendixes: Command definitions, specifications, certifications, terminology definitions, licenses, service and warranty details.

The most recent version of this manual is always at www.blackbox.com.

TYPES OF USERS

The console server supports two classes of users:

1. First, there are administrative users, who have unlimited configuration and management privileges over the console server and all the connected devices.

Administrative users are set up as members of the admin user group. Users in this class are referred to in this manual as Administrators. An Administrator can access and control the console server using the config utility, the Linux command line or the browser-based Management Console. By default, the Administrator has access to all services and ports to control all the serial connected devices and network connected devices (hosts).

2. The second class of users embraces those who have been set up by an Administrator with specific limits of their access and control authority. These users are set up as members of one of the pre-configured user groups (pptpd, dialin, ftp, pmshell or users) or another user group an Administrator has added.

They are only authorized to perform specified controls on specific connected devices and are referred to as Users. These Users (when authorized) can access serial or network connected devices and control these devices using the specified services (eg Telnet, HHTPS, RDP, IPMI, Serial-over-LAN, Power Control).

An authorized User also has a limited view the Management Console and can only access authorized configured devices and review port logs.

In this manual, when the term user (lower case) is used, it is referring to both classes of users above. This document also uses the term remote users to describe users who are not on the same LAN segment as the console server.

These remote users may be users who are on the road connecting to managed devices over the public Internet. They may be an Administrator in another office connecting to the console server itself over the enterprise VPN. Or the remote user may be in the same room or the same office but connected on a separate VLAN to the console server.

MANAGEMENT CONSOLE

The features of your console server are configured and monitored using the Black Box Management Console. When you first browse to the Management Console, you can use the menu displayed on the left side to configure the console server. Once you have completed the initial configuration, you can continue to use the Management Console. It runs in a browser and provides a view of the console server and all the connected devices.

Administrators can use the Management Console, either locally or from a remote location, to configure and manage the console server, users, ports, hosts, power devices and associated logs and alerts.

Users can also use the Management Console, but have limited menu access to control select devices, review their logs, access them using the Web terminal, or control power to them.

The console server runs an embedded Linux operating system, and experienced Linux and UNIX users may prefer to undertake configuration at the command line.

You can gain command line access by cellular, dial-in, or by directly connecting to the console server's serial console port (aka the console server's modem port). The shell can also be accessed via ssh or Telnet over a LAN or by connecting with PPTP, IPsec or OpenVPN.

ABOUT THIS MANUAL

WHERE TO FIND ADDITIONAL INFORMATION

- The Quick Start Guide that came with your console server.

CHAPTER 1: SPECIFICATIONS

SPECIFICATIONS: LES1500 SERIES CONSOLE SERVERS

Console Specifications	
Console Ports	LES1516A: (16) RJ-45 RS-232 serial ports with Cisco pinouts; LES1532A: (32) RJ-45 RS-232 serial ports with Cisco pinouts; LES1548A: (48) RJ-45 RS-232 serial ports with Cisco pinouts
Interface	
Ethernet Ports	(2) 10-/100-/1000-Mbps Ethernet RJ-45 ports
Console Port	(1) DB9 RS-232 console port
Serial Ports	Software-selectable, 50 to 230,400 bps
USB	(2) USB 2.0 ports for increased storage
Remote Access	Dual Ethernet, aggregation and redundancy, remote access automatic network failover, easy browser UI IPv6
Console Management	Built-in web terminal, SSH direct to consoles, optional console keystroke logging, alert on cable disconnects, text pattern match and more, inline power control, multiple concurrent sessions
Power Requirements	
Power Supply	LES1508A: External AC/DC power supply; LES1516A, LES1532A, LES1548A: Single AC power supply
Power Consumption	Less than 30 W
Physical	
Dimensions	1.75"H x 17"W x 6.9"D (4.5 x 43.2 x 4.5 cm)
Weight	9 lb. (4 kg)
Form Factor	1 RU
Memory and CPU	
CPU	Marvell 88F6W11
Memory	32 MB Flash
Internal Storage	4 GB
Environmental	
Operating Temperature	41 to 122° F (5 to 40° C)
Storage Temperature	-20 to +140° F (-30 to +60° C)
Humidity	5 to 90%
Security, Encryption and Authentication	SSH; FIPS-140-2 compliant; Open SSL Module; Strong ciphers—AES encryption; IPsec; AAA, TACACS+, RADIUS, Active Directory, OpenLDAP, Kerberos, with local fallback; Two factor authentication via remote AAA; Configurable stateful firewall; OpenVPN

SPECIFICATIONS (CONTINUED): LES1500 SERIES CONSOLE SERVERS

Automation and Scalability	ZTP, Virtual Central Management System (VCMS); RESTful API, programmable and extensible; Auto-Response, SNMP, LLDP, NTP
Certifications	
Emissions	FCC Part 15 Subpart B Class A; ICES-003 Issue 4 February 2004; AS/NZS CISPR 22: 2004 Class A; EN 55022 Emissions Class A (2009) A1 (2010); EN 61000-3-2 Harmonics Current Emissions (2014); EN 61000-3-3 Voltage Fluctuation and Flicker (2013)
Immunity	EN 55024 ITE Immunity (2010); IEC 61000-4-2 (2008); IEC 61000-4-3 (2006) A1 (2007) A2 (2010); IEC 61000-4-4 (2012); IEC 61000-4-5 (2014); IEC 61000-4-6 (2013); IEC 61000-4-8 (2009); IEC 61000-4-11 (2004)
Other Agency Approvals	CE, UL 1950, TUV, C-Tick, RoHS compliant, Security features to support NERC CIP standards, FIPS 140-2 validated module Certificate #2473, CCC
MTBF	150,000 hours

SPECIFICATIONS: LES1600 CONSOLE SERVERS

Console Specifications	
Console Ports	LES1604A, LES1604A-R-R2: (4) RJ-45 RS-232 Cisco straight pinout console ports; LES1608A: (8) RJ-45 RS-232 Cisco straight pinout console ports
Interface	
Ethernet Ports	(2) 10-/100-/1000-Mbps Ethernet RJ-45 ports
Serial Ports	Cisco straight 50 to 230,400 bps
USB Ports	(4) USB 2.0 console ports
Remote Access	All LES1600 models: Dual Ethernet, aggregation and redundancy, Automatic Network Failover, Easy browser UI, IPv6; LES1604A-R-R2: 4G Cellular
I/O Ports	(2) TTL level digital I/O ports (DIO), 5 VDC max. @ 20 mA (supports external water/smoke/motion dry contact sensors); (2) high-voltage digital outputs (HVDO), 5–30 VDC @100 mA (drives relays, alarms, etc.)
Console Management	Built-in web terminal, SSH direct to consoles, optional console keystroke logging, alert on cable disconnects, inline power control, multiple concurrent sessions
Power Requirements	
Power Adapter	110–240 VAC to 12 VDC external power adapter
Power Consumption	Less than 11.5 W
Physical	
Dimensions	5.1"H x 4.8"W x 1.4"D (13 x 12 x 3.5 cm)
Form Factor	Compact
Memory and CPU	
CPU	800 MHz Armada 370 ARMv7 SoC (Marvell 88F6W11)
Memory	256 MB DDR3 SDRAM; 32 MB Embedded NOR Flash
Internal Storage	4 GB NAND Flash
Environmental	
Operating Temperature	-13 to +140° F (-25 to +60° C)
Storage Temperature	-40 to +167° F (-40 to +75° C)
Humidity	5 to 90%
Security, Encryption and Authentication	SSH; FIPS-140-2 compliant Open SSL Module; Strong ciphers—AES encryption; Cisco-compatible IPsec; AAA—TACACS+, RADIUS, Active Directory/OpenLDAP, Kerberos, with local fallback; Two factor authentication via remote AAA; Configurable stateful firewall; OpenVPN

SPECIFICATIONS: LES1600 CONSOLE SERVERS (CONTINUED)

Automation and Scalability	ZTP, Virtual Central Management System (VCMS); RESTful API, programmable and extensible; Auto-Response, SNMP, LLDP, NTP
Certifications	
Emissions	FCC Part 15 Subpart B:2015; EN55022:2010; CISPR 22:2008; ICES-003 Issue 5 (2014); AS/NZS CISPR 22: 2009+ A1:2010; EN 61000-3-2:2006/A2:2009; EN 61000-3-3:2008
Immunity	EN 55024:2010 CISPR 24:2010; EN 61000-4-2:2009; EN 61000-4-3:2006+A2:2010; EN 61000-4-4:2004+A1:2010; EN 61000-4-5:2006; EN 61000-4-6:2009; EN 61000-4-8:2010; EN 61000-4-11:2004
Other Agency Approvals	CE, UL 1950, TUV, C-Tick, RoHS compliant, Security features to support NERC CIP standards, FIPS 140-2 validated module Certificate #2473, CCC

SPECIFICATIONS: LES1700-R2 SERIES CONSOLE SERVERS

Console Specifications	
Console Ports	LES1708A-R2: (8) RJ-45 RS-232 software-selectable console ports; LES1716A-R2: (16) RJ-45 RS-232 software-selectable console ports; LES1732A-R2: (32) RJ-45 RS-232 software-selectable console ports; LES1748A-R2: (48) RJ-45 RS-232 software-selectable console ports
Interface	
Ethernet Ports	(2) 10-/100-/1000-Mbps Ethernet/SFP fiber ports with 1500 VAC isolation and ESD protection
Console Port	(1) RJ-45 RS-232 console port
PSTN Modem	(1) internal V.92 modem with RJ-11 socket
Serial Ports	Software-selectable, 50 to 230,400 bps
USB	(2) USB 3.0 ports for increased storage
Remote Access	Integrated V.92 PSTN dial-in, dual Ethernet, aggregation and redundancy, remote access automatic network failover, easy browser UI IPv6
Console Management	Built-in web terminal, SSH direct to consoles, optional console keystroke logging, alert on cable disconnects, text pattern match and more, inline power control, multiple concurrent sessions
Power Requirements	
Dual AC	Dual socket, universal 100–240 VAC
Power Consumption	Less than 30 W
Physical	
Dimensions	1.75"H x 17"W x 10"D (4.5 x 44 x 25.4 cm)
Weight	10 lb. (4.5 kg)
Form Factor	1 RU
Memory and CPU	
CPU	1 GHz ARM SoC (Marvell 88F6283)
Memory	256 MB DDR2 SDRAM; 64 MB Embedded NOR Flash
Internal Storage	16 GB
Environmental	
Operating Temperature	41 to 122° F (5 to 40° C)
Storage Temperature	-20 to +140° F (-30 to +60° C)
Humidity	5 to 90%
Environmental Monitoring	Serial EMD5000 to support physical, smoke, water leak and vibration sensors
Security, Encryption and Authentication	SSH; FIPS-140-2 compliant Open SSL Module; Strong ciphers—AES encryption; Cisco-compatible IPsec; AAA—TACACS+, RADIUS, Active Directory/OpenLDAP, Kerberos, with local fallback; Two factor authentication via remote AAA; Configurable stateful firewall; OpenVPN

SPECIFICATIONS (CONTINUED): LES1700-R2 SERIES CONSOLE SERVERS

Automation and Scalability	ZTP, Virtual Central Management System (VCMS); RESTful API, programmable and extensible; Auto-Response, SNMP, LLDP, NTP
Certifications	
Emissions	FCC Part 15 Subpart B Class A; ICES-003 Issue 4 February 2004; AS/NZS CISPR 22: 2004 Class A; EN 55022 Emissions Class A (2009) A1 (2010); EN 61000-3-2 Harmonics Current Emissions (2014); EN 61000-3-3 Voltage Fluctuation and Flicker (2013)
Immunity	EN 55024 ITE Immunity (2010); IEC 61000-4-2 (2008); IEC 61000-4-3 (2006) A1 (2007) A2 (2010); IEC 61000-4-4 (2012); IEC 61000-4-5 (2014); IEC 61000-4-6 (2013); IEC 61000-4-8 (2009); IEC 61000-4-11 (2004)
Other Agency Approvals	CE, UL 1950, TUV, C-Tick, RoHS compliant, Security features to support NERC CIP standards, FIPS 140-2 validated module Certificate #2473, CCC
MTBF	150,000 hours

2.1 AVAILABLE MODELS COMPARISON CHARTS

TABLE 2-1. AVAILABLE MODELS COMPARISON CHART

PRODUCT CODE	SERIAL RS-232	USB 2.0	USB 3.0	NETWORK 10/100/1000	FLASH	RAM	INTERNAL MODEM	WIRELESS	POWER
LES1516A	16	2	—	2	32 MB	4 GB	—	—	Single AC
LES1532A	32	2	—	2	32 MB	4 GB	—	—	Single AC
LES1548A	48	2	—	2	32 MB	4 GB	—	—	Single AC
LES1604A	4	4	—	2	256/32 MB	4 GB	—	—	Single AC
LES1604A-R-R2	4	4	—	2	256/32 MB	4 GB	cellular 4G	—	Single AC
LES1608A	8	4	—	2	256/32 MB	4 GB	—	—	Single AC
LES1708A-R2	8	—	2	2	256/64 MB	16 GB	POTS	—	Dual AC
LES1716A-R2	16	—	2	2	256/64 MB	16 GB	POTS	—	Dual AC
LES1732A-R2	32	—	2	2	256/64 MB	16 GB	POTS	—	Dual AC
LES1748A-R2	48	—	2	2	256/64 MB	16 GB	POTS	—	Dual AC

TABLE 2-2. SOFTWARE FEATURES SUPPORTED PER MODEL SERIES

SERIES	DHCP	DDNS	MGT LAN	CELL OR WI-FI	OOB	AUTO-RESPONSE	FLASH (FTP & TFTP)	FTP S	IPSEC, PPTP AND OPENVPN
LES1600	yes	yes	yes	yes[1]	yes	yes	yes	yes	yes
LES1700-R2	yes	yes	yes	no	yes	yes	yes	yes	yes
LES1516/32/48	yes	yes	yes	no	yes	yes	yes	yes	yes

1. Selected models have 3G/4G cellular.

CAUTION: To avoid physical and electrical hazards, read the Safety Precautions at the beginning of this manual.

2.2 WHAT'S INCLUDED

Your package should include the following items. If anything is missing or damaged, contact Black Box Technical Support at 877-877-2269 or info@blackbox.com

2.2.1 LES1516A, LES1532A, LES1548A

- (1) Console Server
- (2) CAT5 UTP cables
- (1) DB9F to RJ-45 straight-through adapter
- (1) DB9F to RJ-45 crossover adapter
- (1) IEC AC power cord
- (1) Rackmount Kit: (2) Brackets, (4) 10/32 Screws, (6) 6/32 Screws, (4) Cage Nuts
- (1) Quick Start Guide

2.2.2 LES1600 SERIES

- (1) LES1604A, LES1604A-R-R2, or LES1608A Console Server
- (1) DB9F-to-RJ-45 crossover serial adapter
- (1) 12-VDC switching DC power supply with US, UK, EU, AU adapters
- (1) Rackmount Kit: (2) Brackets, (4) 10/32 Screws, (6) 6/32 Screws, (4) Cage Nuts
- (4) Adhesive-backed rubber feet
- (1) Digital I/O converter (terminal block)
- (1) SIM holder
- (2) Antennas
- (1) Quick Start Guide

2.2.3 LES1700-R2 SERIES

- (1) LES1708A-R2, LES1716A-R2, LES1732A-R2 or LES1748A-R2 Console Server
- (1) Rackmount Kit: (2) Brackets, (4) 10/32 Screws, (6) 6/32 Screws, (4) Cage Nuts
- (2) Power cords
- (2) 6-ft. (1.8-m) CAT5 patch cables
- (4) DB9F to RJ-45 adapters
- (2) DB9M to RJ-45 adapters
- (2) Loopback connectors RJ-45
- (1) Quick Start Guide

2.3 HARDWARE DESCRIPTION

While we cannot illustrate every possible model of the Console Server in this manual, Sections 2.3.1 through 2.3.3 show one model from each series.

2.3.1 LES1500 SERIES

Figures 2-1 and 2-2 show the front and back panels of the LES1548A. Table 2-3 describes its components.

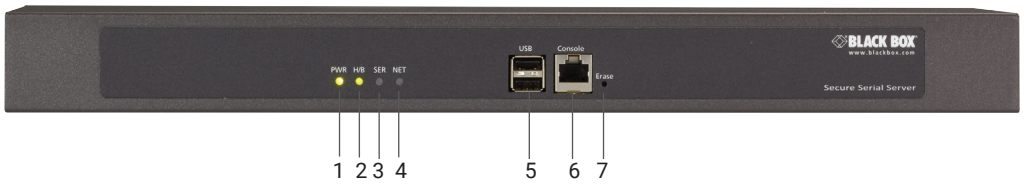


FIGURE 2-1. LES1548A FRONT PANEL

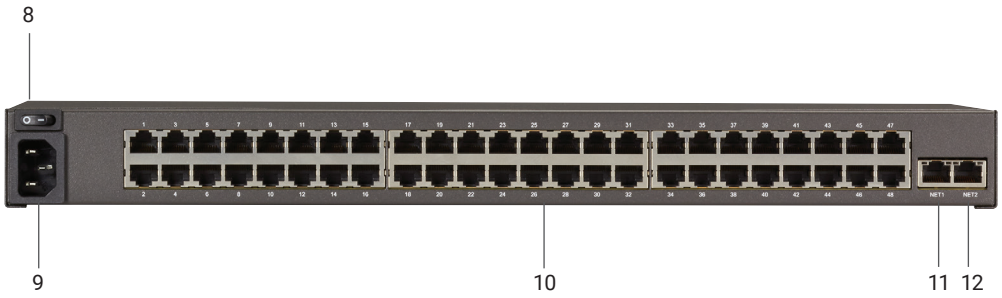


FIGURE 2-2. LES1548A BACK PANEL

TABLE 2-3. LES1548A CONSOLE SERVER COMPONENTS

NUMBER IN FIGURE 2-1 OR 2-2	COMPONENT	DESCRIPTION
1	(1) PWR LED	Lights when power is on
2	(1) H/B LED	Used for flash firmware updates
3	(1) SER LED	Serial connection indication
4	(1) NET LED	Links to Network 1
5	(2) USB ports	Allow attachment of peripherals such as additional storage and USB consoles
6	(1) RJ-45 console port	Links to RS-232 console
7	Erase button	Push to erase settings
8	(1) I/O switch	Press to turn power ON or OFF
9	(1) 3-prong power receptacle	Links to power supply
10	(48) RJ-45 serial ports	Links to devices
11	(1) RJ-45 port	NET1
12	(1) RJ-45 port	NET2

2.3.2 LES1600 SERIES

Figures 2-3 and 2-4 show the front and back panels of the LES1604A. Table 2-4 describes its components.

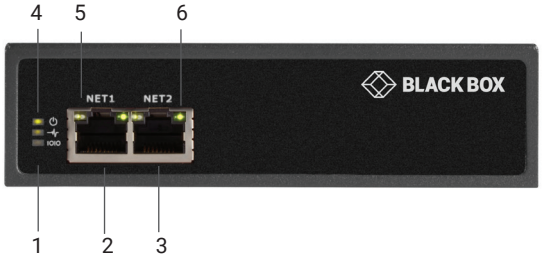


FIGURE 2-3. LES1604A FRONT PANEL

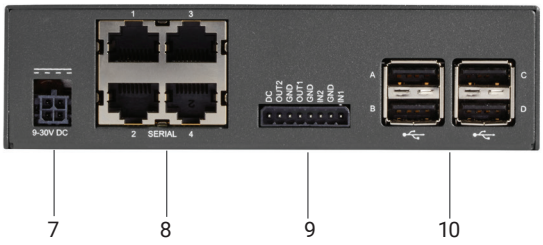


FIGURE 2-4. LES1604A BACK PANEL

TABLE 2-4. LES1604A CONSOLE SERVER COMPONENTS

NUMBER IN FIGURE 2-3 OR 2-4	COMPONENT	DESCRIPTION
1	H/B LED	Heartbeat LED, lights when firmware is running
1	Serial LED	Active serial communication
2	(1) RJ-45 port	NET1 Ethernet
3	(1) RJ-45 port	NET2 Ethernet
4	(1) Power LED	Lights when power is on
5, 6	(1) Speed and Activity LEDs	Indicates 10/100/1000 Mbps and activity on NET1 and NET2
7	(1) 4-pin power port	Power adapter input
8	(4) RJ-45 ports	Serial console ports
9	DIO and HVDO ports	DIO and HVDO ports
10	(4) USB ports	Link to USB consoles

2.3.3 LES1700-R2 SERIES

Figures 2-5 and 2-6 show the front and back panels of the LES1716A-R2. Table 2-5 describes its components.

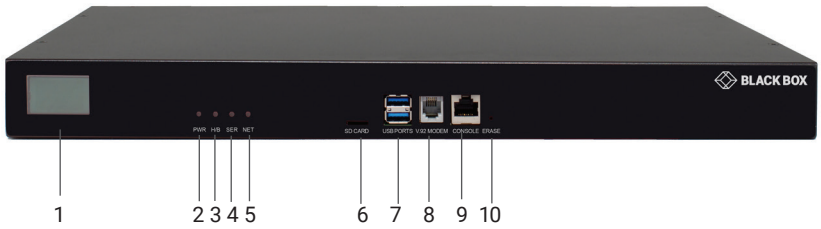


FIGURE 2-5. LES1716A-R2 FRONT PANEL

FIGURE 2-6. LES1716A-R2 BACK PANEL

TABLE 2-5. LES1716A-R2 CONSOLE SERVER COMPONENTS

NUMBER IN FIGURE 2-5 OR 2-6	COMPONENT	DESCRIPTION
1	LCD screen	LCD configuration
2	PWR LED	Lights when power is on
3	H/B LED	Heartbeat LED, lights when firmware is running
4	Serial LED	Active serial communication
5	NET LED	Active network communication
6	SD card slot	Expand storage
7	(2) USB ports	Link to USB consoles
8	V.92 modem port	Port for analog dialup
9	RJ-45 console port	Attaches to serial console
10	Erase button	Push to erase settings and configuration
11, 12	(2) 3-prong power receptacles	Link to power supplies
13	(2) I/O switches	Turns power ON or OFF
14	(16) RJ-45 ports	Link to serial devices
15, 16	(2) RJ-45/SFP ports	Link to Ethernet network ports NET1 and NET2

CHAPTER 3: INSTALLATION

Connect the Console Server to the network, to the serial ports of the controlled devices, and to power as outlined below.

3.1 POWER CONNECTION

3.1.1 LES1700-R2, LES1516A, LES1532A AND LES1548A MODELS

These standard LES1700-R2 console servers have dual universal AC power supplies with auto failover built in. The power supplies accept AC input voltage between 100 and 240 VAC with a frequency of 50 or 60 Hz and the total power consumption per console server is less than 30 W. The LES1516A, LES1532A and LES1548A console servers each have one AC power supply.

Two IEC AC power sockets are located at the rear of the metal case, and these IEC power inlets use conventional IEC AC power cords. Power cords for various regions are available, although the North American power cord is provided by default. There is a warning notice printed on the back of each unit.

NOTE: To avoid electrical shock, the power cord grounding conductor must be connected to ground.

3.1.2 LES1600 MODELS

LES1600 models are supplied with an external AC-12-VDC wallmount power supply. This comes with a selection of wall socket adapters for each geographic region (North American, Europe, UK, Japan or Australia). The 12-VDC connector from the power supply unit plugs into the 12-VDC (PWR) power jack on the side of the console server casing.

- Plug in the power supply AC power cable and the DC power cable.
- Turn on the AC power and confirm the console server Power LED (PWR) is lit.

The LES1600 models can also be powered from an external +9-VDC to +30-VDC power source by connecting the DC power lines to a power plug that plugs into the 12-VDC (PWR) jack.

3.2 NETWORK CONNECTION

All Black Box console servers ship with Ethernet ports.

These ports are located on the rear panel of the rackmount LES1516A, LES1532A, LES1548A units, and on the front of the smaller LES1600 units. All physical connections are made using either industry standard CAT5 cabling and connectors or small form-factor pluggable transceivers (SFPs).

Make sure you only connect the LAN port to an Ethernet network that supports 10/100/1000 Mbps (LES1700-R2, LES1516A, LES1532A, LES1548A, LES1600 only).

The LES1700-R2 has four physical input ports which are logically presented as two ports (NET1 and NET2). Each logical port consists of a copper 10/100/1000 port and a fiber-optic small form-factor pluggable (SFP) module slot.

The LES1604A has six physical input ports: (2) RJ-45 copper ports on the front of the device which are logically paired and marked as NET1 and NET2; and four RJ-45 ports on the back of the device which constitute an independent Ethernet switch. The LES1608A has ten physical RJ-45 input ports: two RJ-45 ports on the front of the device and eight RJ-45 ports on the back of the device.

For LES1700-R2 series devices with logically-paired SFP and RJ-45 ports, you can use only one of the two physical ports at a time: either the SFP module port or the 10/100/1000 port.

For LES1700-R2 console servers with logically-paired SFP and RJ-45 ports, the fiberoptic medium (the SFP module) has priority over the copper medium (the RJ-45 port). Only if the SFP module is not plugged in does the RJ-45 copper link become active. This applies regardless of the connection order. If the SFP module is plugged in after the copper medium has established a link, the copper link is disconnected and the fiberoptic medium becomes active.

For the initial configuration of the console server, you must connect a computer to the console server's principal network port. This port's label varies from model to model but always includes a numeric one (1). Specific labels include NET1, NETWORK1, LAN1, and LAN USB1.

3.3 SERIAL PORT CONNECTION

Console servers all come with four to forty eight serial ports, marked SERIAL or SERIAL PORTS. These ports connect to serially Managed Devices. Each console server also has either a dedicated Local Console (or modem) port marked LOCAL or CONSOLE, or one of its SERIAL ports can be software configured in Local Console mode. This Local Console port can be used for local command-line access (or an external serial modem out-of-band connection).

All console server models except the LES1600 have a dedicated local RS-232 Console port. This is an RJ-45 connector (Cisco Straight) located on the front of the LES1700-R2, LES1516A, LES1532A and LES1548A models.

LES1600 models have four or eight serial ports presented as RJ-45 ports 1–x. By default, port 1 on all these models is configured in Local Console mode.

Conventional CAT5 cabling with RJ-45 jacks is generally used for serial connections. Black Box supplies a range of cables and adapters that may be required to connect to the more popular servers and network appliances.

Before connecting the console port of an external device to the console server serial port, confirm that the device supports the RS-232C (EIA-232) standard.

The console servers come with four to forty eight serial connectors for the RS-232 serial ports:

The RJ-45 serial ports are located on the rear panel of the LES1600 and on the rear panel of the rackmount LES1700-R2.

The LES1600, LES1516A, LES1532A, and LES1548A models have Cisco Straight serial pinouts on the RJ-45 connectors.

All serial ports on the LES1700-R2 are RJ-45 and are software-selectable for Cisco Straight or Cisco Rolled pinout.

Some console server models support RS-422 and RS-485 as well as RS-232.

The four RJ-45 serial ports on the LES1604A are each RS-232/422/485 software-selectable.

See Appendix C for RS-422/485 pinout and connection details.

TABLE 3-1. SERIAL PORT PINOUTS

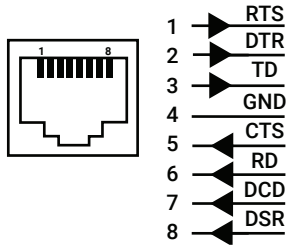
PRODUCT FAMILY	CONNECTOR	SERIAL PORTS PINOUT	RS-232	RS-422/485	CONSOLE PORT
LES1600 series	RJ	X2 Cisco	yes	yes	no[1]
LES1700-R2 series	RJ	X2 Cisco	yes	no	no[1]
LES1500 series	RJ	X2 Cisco	yes	no	yes

[1]NOTE. The first serial port can be reassigned to be a console port.

3.3.1 CISCO ROLLED RJ-45 PINOUT

The LES1700-R2 console servers can select this pinout. This makes it easy to replace Cyclades products, and is convenient for use with rolled RJ-45 cable:

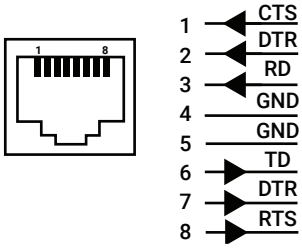
TABLE 3-2. CISCO ROLLED RJ-45 PINOUT

DIAGRAM	PIN	SIGNAL	DEFINITION	DIRECTION
	1	RTS	Request To Send	Output
	2	DTR	Data Terminal Ready	Output
	3	TXD	Transmit Data	Output
	4	GND	Signal Ground	n/a
	5	CTS	Clear To Send	Input
	6	RXD	Receive Data	Input
	7	DCD	Data Carrier Detect	Input
	8	DSR	Data Set Ready	Input

3.3.2 CISCO RJ-45 PINOUT

The LES1600, LES1516A, LES1532A and LES1548A models have Cisco serial pinouts on their RJ-45 connectors. The LES1700-R2 console servers can select this pinout (it is the default). This provides straight-through RJ-45 cable to equipment such as Cisco, Juniper, Sun and many more:

TABLE 3-3. CISCO RJ-45 PINOUT

DIAGRAM	PIN	SIGNAL	DEFINITION	DIRECTION
	1	CTS	Clear To Send	Output
	2	DSR	Data Set Ready	Output
	3	RXD	Receive Data	Output
	4	GND	Signal Ground	n/a
	5	GND	Signal Ground	Input
	6	TD	Transmit Data	Input
	7	DTR	Data Terminal Detect	Input
	8	RTS	Request To Send	Input

3.4 USB PORT CONNECTION

Most console servers have external USB ports. LES1700-R2 Series Console Servers have USB 3.0 ports. On other models, these ports are mostly USB 2.0. They can be used for:

- connecting to UPS or PDU managed devices (for managing UPS supplies, for example).
- connecting an external USB memory stick
- connecting to USB Consoles.

The LES1700-R2 series models have two front-facing USB 3.0 ports.

Some console server models also come with internal USB connections to cellular modem and/or flash memory.

The LES1600 models have an internal 4 GB USB flash drive as well as four unallocated external USB 2.0 ports. These four unallocated USB ports are labelled 1–4 on the device itself and in the Web interface.

3.5 FITTING CELLULAR SIM AND ANTENNAS

3.5.1 LES1604A-R-R2 MODEL

LES1600s come with internal 4G LTE modems and dual mini-SIM card slots (LES1604A-R-R2).

The LES1604A-R-R2 works with AT&T USA, Verizon USA, and global 4G LTE carriers.

NOTE: The LES1604A-R-R2 model is multi-carrier.

Whichever carrier you choose, their SIM card activates the data plan and must be installed before powering on the device.

Dual-SIM models use a SIM cradle. The cradle holds the SIM card or cards and slides into the dual-SIM-card slot on the front of the device. The bottom slot is the default slot. If you have a dual-SIM LES1600 and only one SIM card, insert the card into the bottom slot of the SIM cradle. No matter the specific configuration, SIM cards go into the cradle with the contacts upwards and the notch inward and adjacent to the longer cradle arm.

LES1600 models also come with two external 7-band cellular antennas. Screw the provided antennas on to the main Cell (M) and diversity Cell (A) SMA connectors on the rear panel. An external GPS passive antenna with magnetic base, SMA connector and 2 meter cable is available (not included). Screw it on to the GPS SMA connector on the rear panel.

3.5.2 ALL LES1700-R2 MODELS

The LES1700-R2 models do not have a cellular modem and do not support Wi-Fi.

CHAPTER 4: SYSTEM CONFIGURATION

This page provides an overview of the current users. Currently the only way to login as another user on the web server is to close and This chapter provides step-by-step instructions for the initial configuration of your console server, and connecting it to the Management or Operational LAN. This involves the Administrator:

- activating the Management Console.
- changing the Administrator password.
- setting the IP address console server's principal LAN port.
- selecting the services to be enabled and access privileges.

This chapter also discusses the communications software tools that the Administrator may use in accessing the console server, and the configuration of the additional LAN ports.

NOTE: For guidance on configuring large numbers of Black Box appliances and/or automating provisioning, consult Section 4.7: Configuration over DHCP (ZTP) and Section 16.15: Bulk Provisioning.

4.1 MANAGEMENT CONSOLE CONNECTION

Your console server comes configured with the following default IP address and subnet mask:

- IP address: 192.168.0.1
- Subnet mask: 255.255.255.0

For initial configuration, we recommend that you connect the console server directly to a single computer.

If you choose to connect the console server and computer to a LAN before completing the initial setup steps, the following conditions must be met:

- there must be no other devices on the LAN at IP address 192.168.0.1.
- the console server and the computer must be on the same LAN segment, with no interposed router appliances.

4.1.1 CONNECTED COMPUTER SETUP

To configure the console server with a browser, the connected PC/workstation should have an IP address in the same range as the console server (for example, 192.168.0.100):

To configure the IP address of a computer running Linux, macOS, or Unix:

- run ifconfig.

To configure the IP address of a computer running Windows:

- Click Start -> (Settings ->) Click Start -> Control Panel -> Network and Sharing Center -> Change Adapter Settings.
- Right-click on Local Area Connection and select Properties.
- Select Internet Protocol (TCP/IP) and click Properties.
- Select Use the following IP address and enter the following details:

  IP address: 192.168.0.100

  Subnet mask: 255.255.255.0

- If you want to retain your existing IP settings for this network connection, click Advanced and add the above details as a secondary IP connection.

If it is not convenient to change your computer's network address, you can use the ARP-Ping command to reset the console server's IP address. To do this from a computer running Windows:

- Click Start -> Run (or select All Programs > Accessories > Run).
- Type cmd and click OK to bring up the cmd.exe shell prompt.
- Type arp –d to flush the ARP cache.
- Type arp –a to view the current ARP cache (this should be empty).

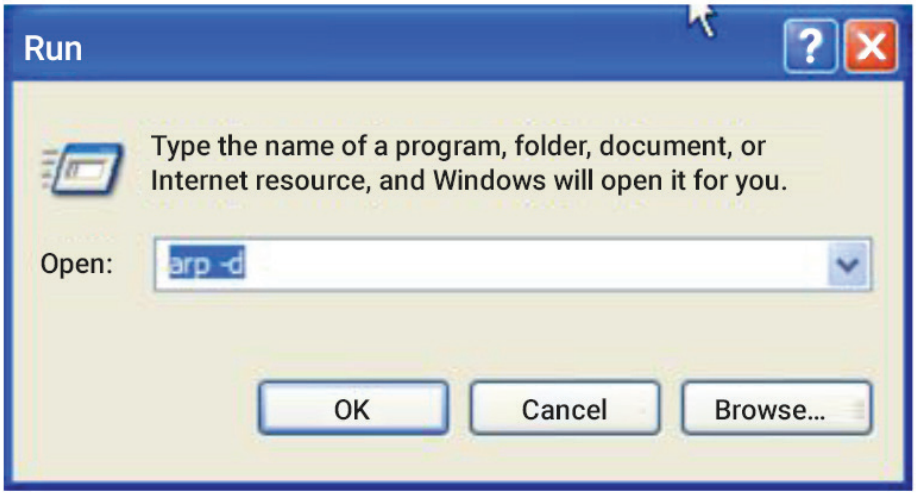


FIGURE 4-1. CMD.EXE SHELL PROMPT

Now add a static entry to the ARP table and ping the console server to assign the IP address to the console server.

In the example below, a console server has the MAC Address 00:13:C6:00:02:0F (designated on the label on the bottom of the unit) and its IP address is set to 192.168.100.23.

NOTE: The computer issuing the arp command must be on the same network segment as the console server (that is, have an IP address of 192.168.100.xxx).

- On Windows: type arp -s 192.168.100.23 00-13-C6-00-02-0F
- On Linux, macOS or Unix: type arp -s 192.168.100.23 00:13:C6:00:02:0F
- Type ping -t 192.18.100.23 to start a continuous ping to the new IP Address.
- Turn on the console server and wait for it to configure itself with the new IP address. It will start replying to the ping at this point.
- Type arp –d to flush the ARP cache again.

4.1.2 BROWSER CONNECTION

Launch or switch to your preferred browser on the connected computer and enter https://192.168.0.1.

NOTE: Console servers ship with a self-signed SSL certificate and are factory configured with HTTPS access enabled and HTTP access disabled.

The Management Console supports all current versions of the popular browsers: Internet Explorer, Firefox, Chrome, Safari and more.

You will be prompted to log in.

Enter the default administration username and administration password:

Username: root

Password: default

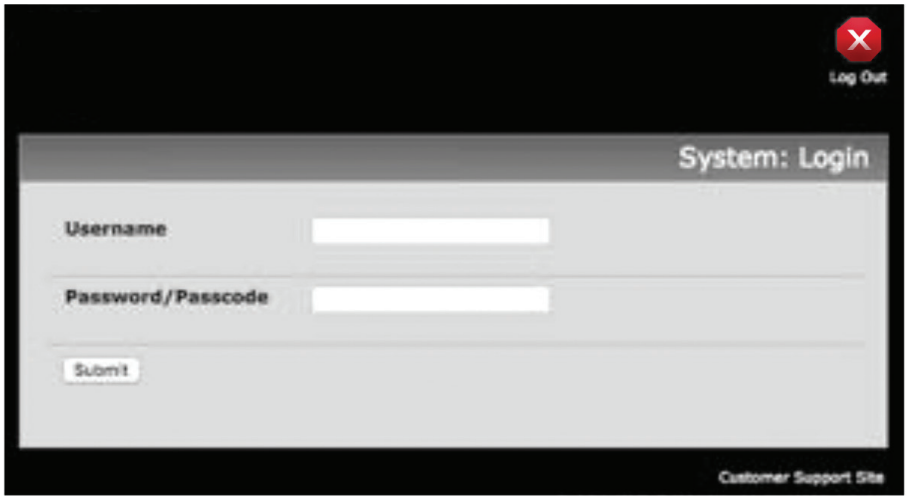


FIGURE 4-2. SYSTEM: LOGIN SCREEN

A Welcome page, which lists initial configuration steps, will display.

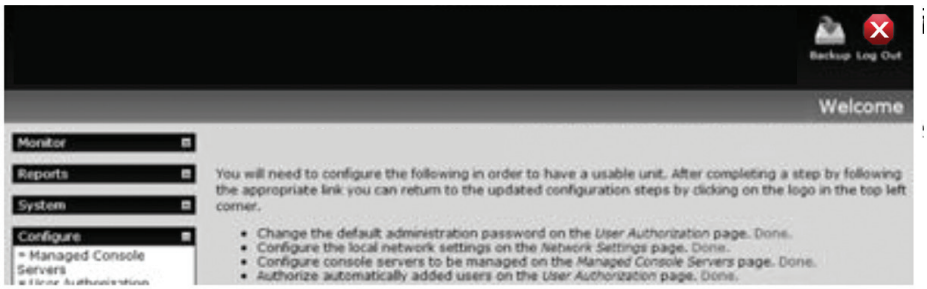


FIGURE 4-3. WELCOME SCREEN

These steps are:

- Change default administration password (Users page, see Section 4.2.)
- Configure the local network settings (System/IP page, see Section 4.3.)
- Configure serial ports settings (Serial & Network/Serial Port page, see Chapter 5.)
- Configure user port access (Serial & Network/Users page, see Chapter 5.)

If your system has a cellular modem, steps to configure the cellular router features will also present:

- Configure the cellular modem connection (System/Dial page, see Chapter 6.)
- Allow forwarding to the cellular destination network (System/Firewall page, see Chapter 6.)
- Enable IP masquerading for cellular connection (System/Firewall page, see Chapter 6.)

After completing each of the above steps, return to the configuration list by clicking the Black Box logo in the top left corner of the page.

NOTE: If you are not able to connect to the Management Console at 192.168.0.1 or if the default Username and Password were not accepted, reset your console server (see Chapter 12).

4.2 ADMINISTRATOR SETUP

4.2.1 CHANGE DEFAULT ROOT SYSTEM PASSWORD

For security reasons, only the administrative user named root can initially log into a console server. So only those people who know the root password can access and reconfigure the console server itself.

The corollary is that anyone who correctly guesses the root password can gain access and control of a console server. The initial root password is default. It is essential, therefore, to enter and confirm a new password before giving the console server any access to, or control of, other computers and network appliances.

- Select Change default administration password from the Welcome page.
- The Serial & Network > Users & Groups page loads. From here a new, confirmed password for the root user can be set.

NOTE: There are no character restrictions in a console server user's password. And passwords can be up to 254 characters long.

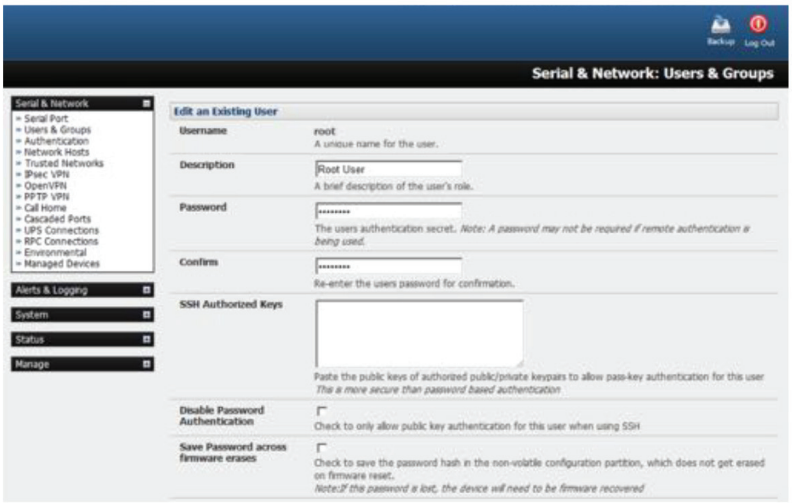


FIGURE 4-4. SERIAL & NETWORK: USERS & GROUPS SCREEN

- If the console server has flash memory (such as the LES1700-R2) you will be given the option to Save Password across firmware erases.
- Checking this will save the password hash in the non-volatile configuration partition, which does not get erased on firmware reset. If this password is lost, the affected console server will need to be firmware recovered.
- Click Apply.

Since the root password has changed, a new log-in prompt will present. This time, use the new password.

4.2.2 SET UP A NEW ADMINISTRATOR

A new Administrator user should be set up and this new user should be used for ongoing console server administration, rather than relying on the root user.

This new user can be configured in the admin group with full access privileges by selecting Serial & Network > Users & Groups >Add a New User.

FIGURE 4-5. ADD A NEW USER SCREEN

4.2.3 NAME THE SYSTEM

- Select System > Administration.
- Enter a System Name and System Description for the console server to give it a unique ID and make it simple to identify.

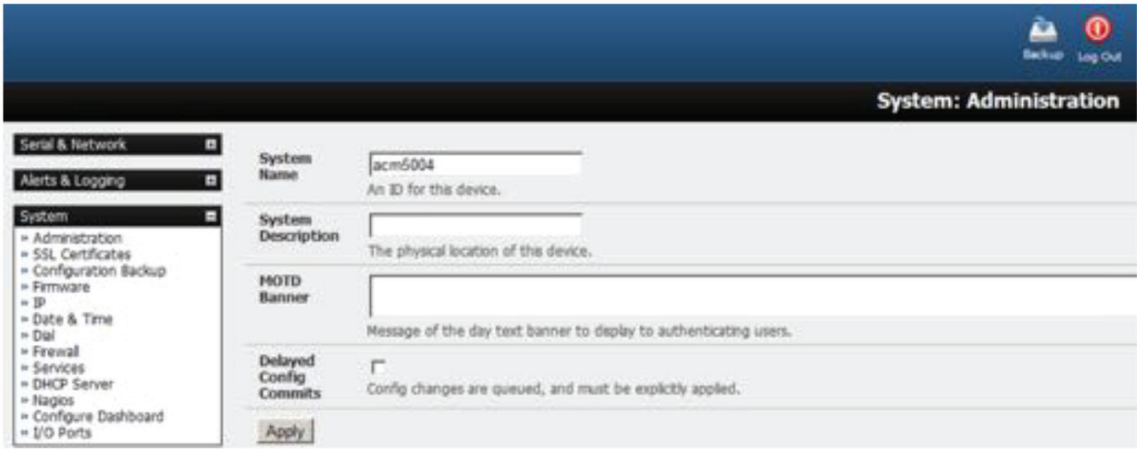


FIGURE 4-6. NAME THE SYSTEM SCREEN

NOTE: The System Name can contain from 1 to 64 alphanumeric characters as well as the following special characters . - _. There are no restrictions on the characters that can be used in the System Description, which can contain up to 254 characters.

- Optional: text entered in the MOTD Banner field is displayed to users when the log-in to the console server.
- Click Apply.

NOTE: If you are not confident your console server has been supplied with the current release of firmware, you can upgrade it. (See Chapter 12 for details.)

4.3 NETWORK CONFIGURATION

The next step is to enter an IP address for the principal Ethernet (LAN/Network/Network1) port on the console server; or enable its DHCP client so that it automatically obtains an IP address from a DHCP server on the network it is to be connected to.

- On the System > IP menu, select the Network Interface page then check DHCP or Static for the Configuration Method.

FIGURE 4-7. NETWORK INTERFACE PAGE

- If you selected Static, you must manually enter the new IP Address, Subnet Mask, Gateway and DNS server details. This selection automatically disables the DHCP client.
- By default, the console server LAN port auto detects the Ethernet connection speed. To lock the Ethernet port to 10 Mbps or 100 Mbps and to Full Duplex (FD) or Half Duplex (HD), select a speed and duplex setting from the Media pop-up menu.

If you encounter packet loss or poor network performance with the default auto-negotiation setting, try manually setting the Media settings on both the console server and the device it is connected to. In most cases, select 100BASE-TX-FD (100 megabits, full duplex). Make sure both sides are set identically.

- If you selected DHCP, the console server will look for configuration details from a DHCP server. This selection automatically disables any static address. The console server's MAC address can be found on a label on the base plate.

In its factory default state (with no Configuration Method selected), the console server has its DHCP client enabled, so it automatically accepts any network IP address assigned by a DHCP server on your network. In this initial state, the console server will then respond to both its Static address (192.168.0.1) and its newly assigned DHCP address.

You may also enter a secondary address or comma-separated list of addresses in CIDR notation as an IP Alias.

For example: 192.168.1.1/24.

NOTE: If you changed the console server's IP address, you may need to reconfigure your computer so it has an IP address that is in the same network range as this new address (as detailed earlier in this chapter).

- Click Apply.
- Reconnect the browser on the computer that is connected to the console server by entering https://new-ip-address-here/.

4.3.1 IPV6 CONFIGURATION

By default, the console server Ethernet interfaces support IPv4. They can also be configured for IPv6 operation.

- Select System > IP.
- Click the General Settings tab.

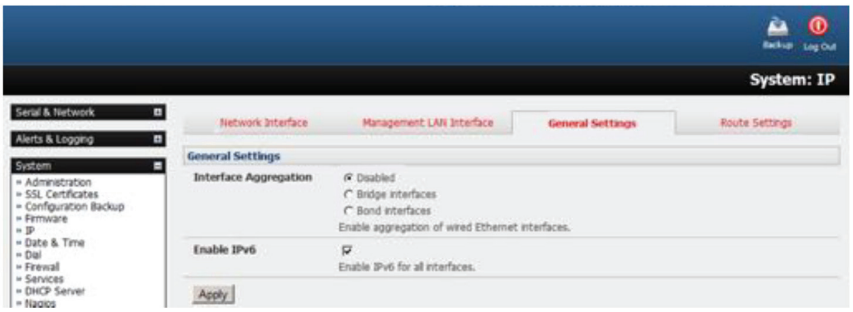


FIGURE 4-8. SYSTEM > IP, GENERAL SETTINGS TAB

- Check the Enable IPv6 check box.

FIGURE 4-9. ENABLE IPV6

- Click the Network Interface to access the IPv6 settings section.
- Configure the IPv6 settings.

4.3.2 DYNAMIC DNS (DDNS) CONFIGURATION

With Dynamic DNS (DDNS), a console server with its IP address dynamically assigned (and that may change from time to time) can be located using a fixed host or domain name.

The first step in enabling DDNS is to create an account with the supported DDNS service provider of your choice. Supported DDNS providers are listed in the following table.

TABLE 4-1. SUPPORTED DDNS SERVICE PROVIDERS

SERVICE PROVIDER	URL	DESCRIPTION
DyNS	http://dyns.cx/	
Dyn	https://dyn.com/	Formerly DynDNS
GNUDip	http://freecode.com/projects/gnudip	An open-source DDNS tool for use by ISPs. Check if your ISP supports GNUDip.
Pubyun	http://pubyun.com/	Chinese DDNS provider. Formerly operated as 3322.org.

NOTE: Two previously supported DDNS providers are ODS, which is no longer operating, and TZO, which was bought by Dyn and is no longer operating independently.

Upon registering with the DDNS service provider, select a username and password, as well as a hostname that you will use as the DNS name (to allow external access to your machine using a URL).

Dynamic DNS service providers allow the user to choose a hostname URL and set an initial IP address to correspond to that hostname URL. Many Dynamic DNS providers offer a selection of URL hostnames available for free use with their service. However, with a paid plan, any URL hostname (including your own registered domain name) can be used.

You can now enable and configure DDNS on any of the Ethernet or cellular network connections on the console server (by default DDNS is disabled on all ports):

- Select the DDNS service provider from the drop down Dynamic DNS list on the System > IP or System > Dial menu.

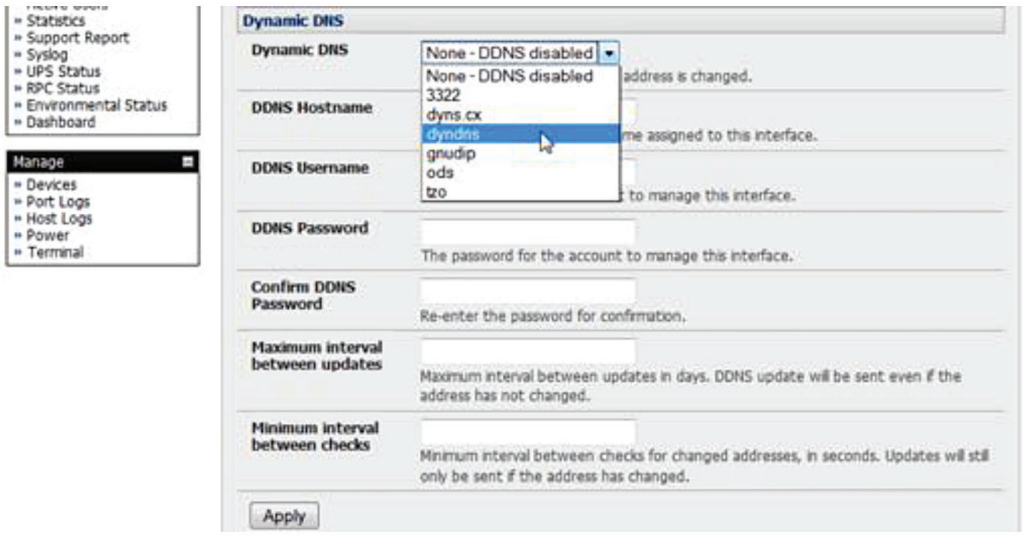


FIGURE 4-10. DROP-DOWN DYNAMIC DNS LIST

- In DDNS Hostname, enter the fully qualified DNS hostname for your console server (for example, your-hostname.dyndns.org).
- •Enter the DDNS Username and DDNS Password for the DDNS service provider account.
- Specify the Maximum interval between updates in days. A DDNS update will be sent even if the address has not changed.
- Specify the Minimum interval between checks for changed addresses in seconds. Updates will still only be sent if the address has changed.
- Specify the Maximum attempts per update (that is, the number of times to attempt an update before giving up). By default this is set to 3.

4.4 SERVICES AND SERVICE ACCESS

The Administrator can access the console server, connected serial ports, and managed devices using a range of access protocols and services. For each such access:

- the particular service must first be configured and enabled to run on the console server.
- then access through the firewall must be enabled for each network connection.

To enable and configure a service:

- Navigate to System > Services.
- Select the Service Settings tab.

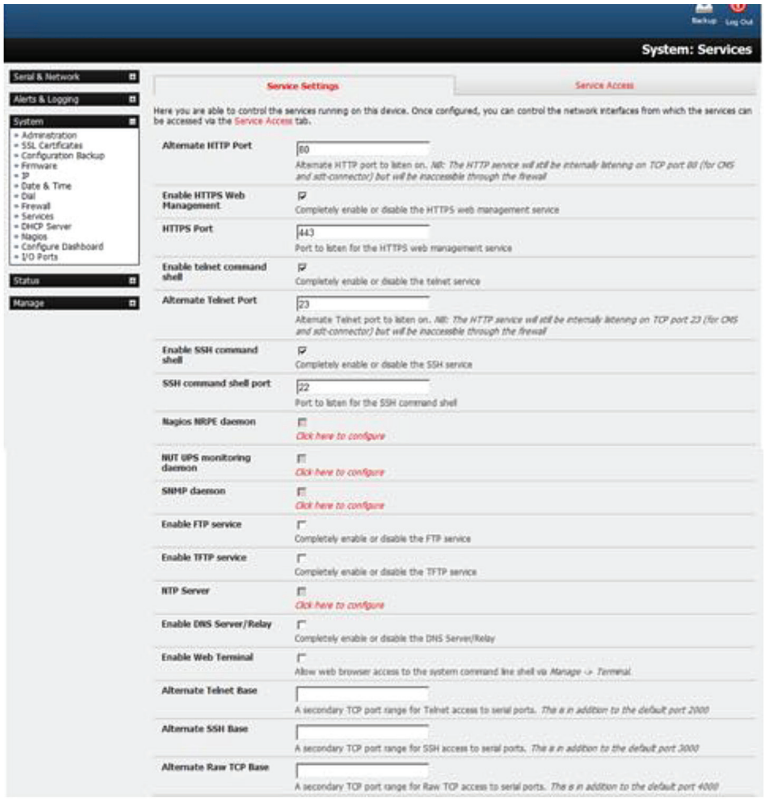


FIGURE 4-11. SYSTEM > SERVICES SCREEN

NOTE: With firmware releases prior to version 3.5.3, services are enabled and configured using the Service Access tab on the System > Firewall page.

Enable and configure basic services.

- HTTP: By default the HTTP service is running and it cannot be fully disabled. However by default HTTP access is disabled on all interfaces and it is recommended this access remains disabled, if the console server is to be remotely accessed over the Internet.

  Alternate HTTP also enables you to configure an alternate HTTP port to listen on. However the HTTP service will continue internally listening on TCP port 80 (for CMS and sdt-connector communications) but will be inaccessible through the firewall.

- HTTPS: By default, the HTTPS service is running and this service is enabled on all network interfaces. We recommend that only HTTPS access be used if the console server is to be managed over any public network (e.g. the Internet). This ensures the Administrator has secure browser access to all the menus on the console server. It also allows appropriately configured Users secure browser access to selected Manage menus. For information on certificate and user client software configuration, see Chapter 10, Authentication. The HTTPS service can be completely disabled (or re-enabled) by checking HTTPS Web Management and an alternate port specified (default port is 443).

- Telnet: By default, the Telnet service is running. However, by default, the service is disabled on all network interfaces.

  Telnet can be used to give the Administrator access to the system command line shell. While this may be suitable for a local direct connection over a management LAN, we recommend that this service be disabled if the console server is to be remotely administered. This service may also be useful for local Administrator and the User access to selected serial consoles.

  The Enable telnet command shell checkbox will completely enable or disable the telnet service. An alternate telnet port to listen on can be specified in Alternate Telnet Port (default port is 23).

- SSH: This service provides secure SSH access to the console server and attached devices —and by default the SSH service is running and enabled on all interfaces. We recommend that you choose SSH as the protocol where the Administrator connects to the console server over the Internet or any other public network. This will provide authenticated communications between the SSH client program on the remote computer and the SSH sever in the console server. For more information on SSH configuration, see Chapter 10, Authentication.

  The Enable SSH command shell checkbox will completely enable or disable this service. An alternate SSH port to listen on can be specified in SSH command shell port (default port is 22).

Enable and configure other services.

- TFTP/FTP: If a USB flash card or internal flash is detected on a console server (for example, an LES1200, LES1508A, LES1600, LES1516A, LES1532A, LES1548A, LES1700-R2 or LES1400) then checking Enable TFTP (FTP) service will enable this service and set up the default tftp and ftp server on the USB flash.

  These servers are used to store config files, maintain access and transaction logs, etc. Files transferred using tftp and ftp will be stored under /var/mnt/storage.usb/tftpboot/ (or /var/mnt/storage.nvlog/tftpboot/ on LES1600-series devices).

  Unchecking Enable TFTP (FTP) service will completely disable the TFTP (FTP) service.

- DNS Relay: Checking Enable DNS Server/Relay will enable the DNS relay feature so clients can be configured with the console server's IP for their DNS server setting, and the console server will forward the DNS queries to the real DNS server.

- Web Terminal: Checking Enable Web Terminal will allow web browser access to the system command line shell via Manage > Terminal.

- Specify alternate port numbers for Raw TCP, direct Telnet/SSH and unauthenticated Telnet/SSH services.

  The console server uses specific default ranges for the TCP/IP ports for the various access services that Users and Administrators can use to access devices attached to serial ports (see Chapter 5: Serial Port, Host, Device and User Configuration). The Administrator can also set alternate ranges for these services, and these secondary ports will then be used in addition to the defaults.

  The default TCP/IP base port address for telnet access is 2000, and the range for telnet is IP Address: Port (2000 + serial port #), that is, ports 2001–2048.

  For example, if the Administrator sets 8000 as a secondary base for telnet, then serial port #2 on the console server can be accessed via telnet at IP Address:2002 and at IP Address:8002.

  The default base for SSH is 3000; for Raw TCP the default base is 4000; and for RFC2217 it is 5000.

A number of other services can be enabled and configured indirectly from this menu by selecting Click here to configure:

- Nagios: Access to the Nagios NRPE monitoring daemons (see Chapter 11).
- NUT: Access to the NUT UPS monitoring daemon (see Chapter 12).
- SNMP: This will enable netsnmp in the console server. SNMP is disabled by default (see Chapter 8 and Section 16.5).
- NTP: See Chapter 12.
- Click Apply. As you apply your services selections, the screen will be updated with a confirmation message: Message Changes to configuration succeeded.

The Services Access settings can now be set to allow or block access.

This specifies which (enabled) services the Administrator can use over each network interface to connect to the console server and, through the console server, to attached serial and network connected devices.

- Navigate to System > Services.
- Select the Service Access tab.

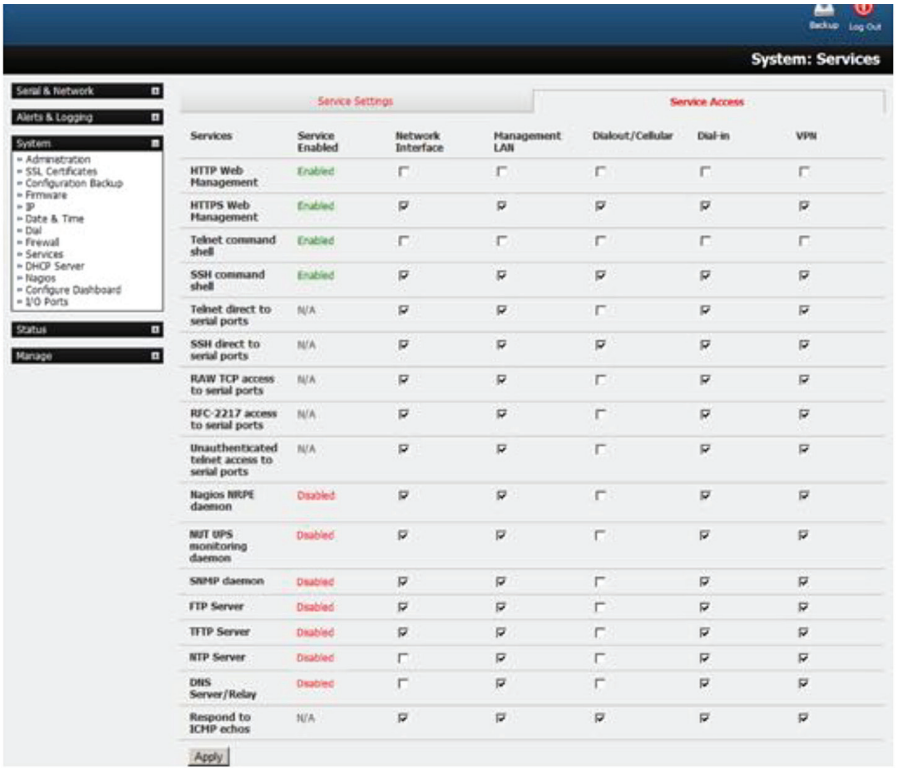


FIGURE 4-12. SERVICE ACCESS TAB

NOTE: With firmware releases pre 3.5.3 the Service Access tab is found at System > Firewall.

The services currently enabled for the console server's network interfaces present. Depending on the particular console server model, the interfaces displayed may include:

- Network interface: for the principal Ethernet connection
- Management LAN/OOB Failover: second Ethernet connections
- Dialout/Cellular: V90 and 3G modem
- Dial-in: internal or external V90 modem

- VPN: IPsec or Open VPN connection over any network interface.

Check or uncheck for each network which service access is to be enabled or disabled.

In the example shown below, local administrators on the local Management LAN have telnet access direct to the console server (and attached serial ports), while remote administrators using Dial-In or Cellular have no telnet access (unless they set up a VPN).

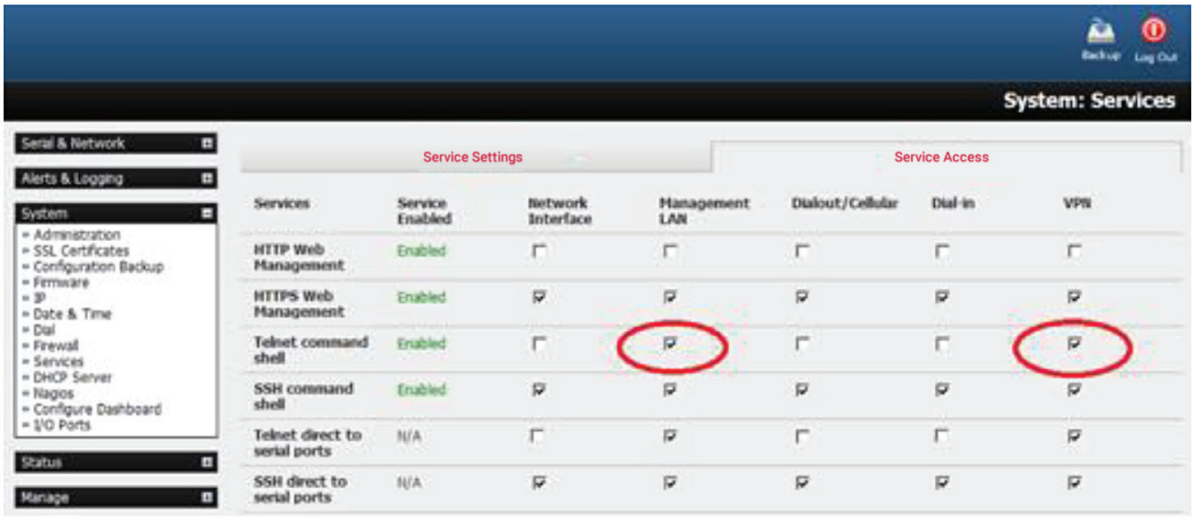


FIGURE 4-13.. SERVICE ACCESS EXAMPLE

The Respond to ICMP echos (that is ping) service access options can be configured at this stage.

This allows the console server to respond to incoming ICMP echo requests. ping is enabled by default. For security reasons, however, this service should generally be disabled post initial configuration.

- You can also configure to allow serial port devices to be accessed from nominated network interfaces using Raw TCP, direct Telnet/SSH, unauthenticated Telnet/SSH services, etc.
- Click Apply to apply your services access selections.

BRUTE FORCE PROTECTION

Brute force protection (Micro Fail2ban) temporarily blocks source IPs that show malicious signs, such as too many password failures.

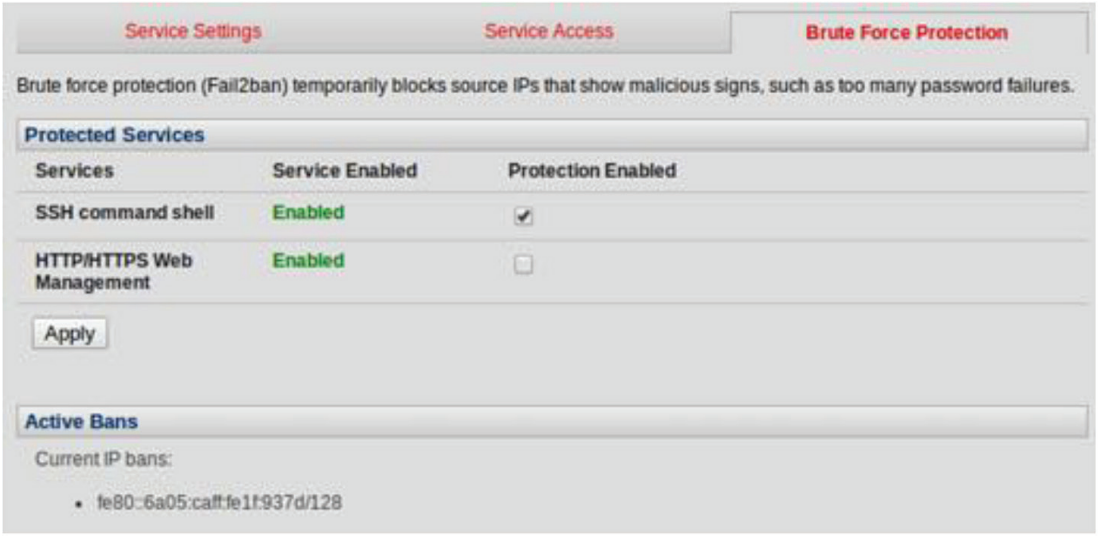


FIGURE 4-14. BRUTE FORCE PROTECTION SCREEN

This may help mitigate scenarios where the Black Box device's network services are exposed to an untrusted network such as the public WAN, and scripted attacks or software worms are attempting to guess (brute force) user credentials and gain unauthorized access.

Brute force protection may be enabled for the listed services.

Once protection is enabled, 3 or more failed connection attempts within 60 seconds from a specific source IP trigger it to be banned from connecting for the next 60 seconds. Active Bans are also listed and may be refreshed by reloading the page.

NOTE: When a Black Box device is running on an untrusted network, we recommend that you use a variety of strategies to lock down remote access. This includes strong passwords (or even better, SSH public key authentication), VPN, and using Firewall Rules to whitelist remote access from trusted source networks only.

4.5 COMMUNICATIONS SOFTWARE

You have configured access protocols for the Administrator client to use when connecting to the console server. User clients (which may be set up later) will also use these protocols when accessing console server serial attached devices and network attached hosts.

You will need to have appropriate communications software tools set up on the Administrator (and User) client's computer. Black Box provides the SDT Connector as the recommended client software tool. Other generic tools such as PuTTY and SSHTerm may be used and these are all described next.

4.5.1 SDT CONNECTOR

SDT Connector is a lightweight tool that enables Users and Administrators to securely access the Console server, and the various computers, network devices and appliances that may be serially or network connected to the console server.

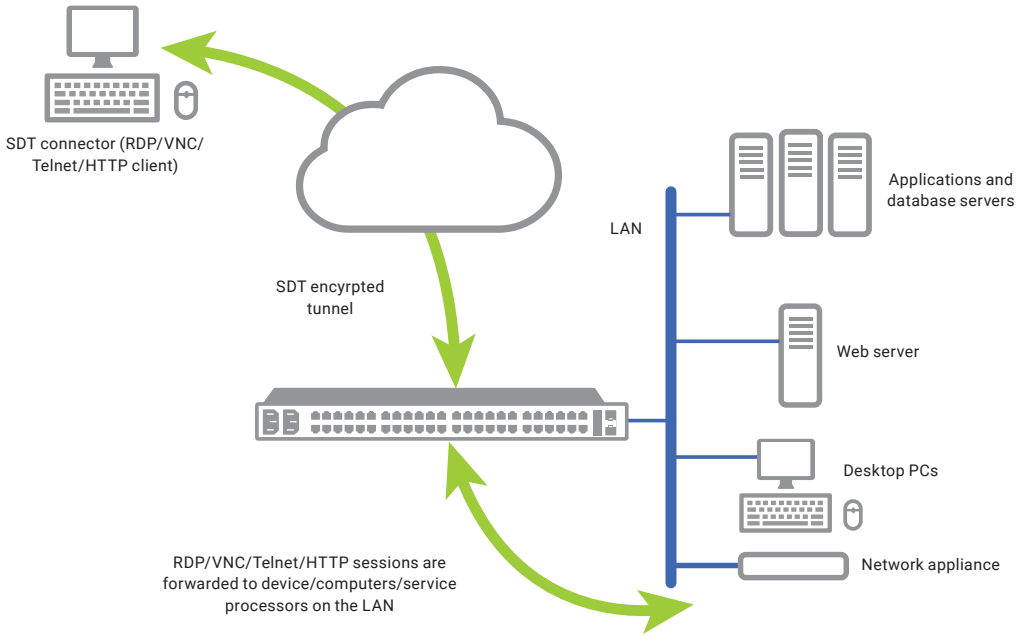


FIGURE 4-15. SDT CONNECTOR APPLICATION EXAMPLE

SDT Connector is a Java client program that couples the trusted SSH tunneling protocol with popular access tools such as Telnet, SSH, HTTP, HTTPS, VNC, and RDP to provide point-and-click secure remote management access to all the managed systems and devices.

Information on using SDT Connector for browser access to the console server's Management Console, Telnet/SSH access to the console server command line, and TCP/UDP connecting to hosts that are network connected to the console server can be found in Chapter 7, SSH Tunnels and SDT Connector.

SDT Connector can be installed on computers running Windows or macOS and on most Linux, UNIX and Solaris systems.

4.5.2 PUTTY

Communications packages like PuTTY can be also used to connect to the Console server command line (and to connect serially attached devices as covered in Chapter 5). PuTTY is a freeware implementation of Telnet and SSH for Win32 and UNIX platforms. It runs as an executable application without needing to be installed onto your system. PuTTY (the Telnet and SSH client itself) can be downloaded from http://putty.org/.

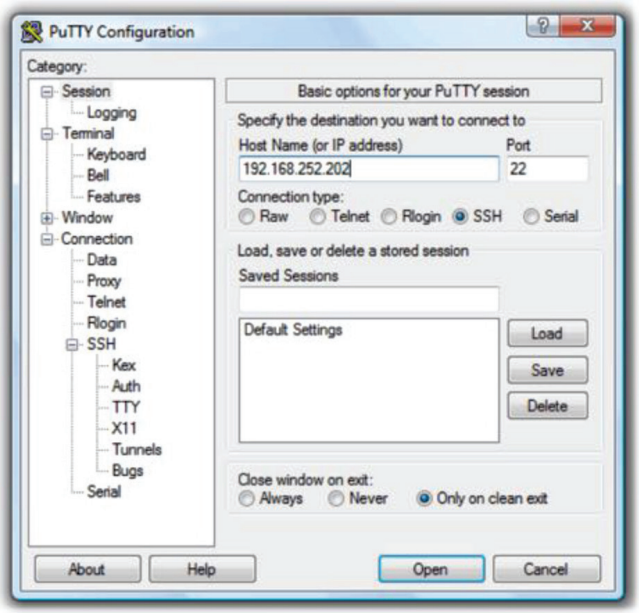


FIGURE 4-16. PUTTY

To use PuTTY for an SSH terminal session from a Windows client, you enter the console server's IP address as the Host Name (or IP address).

To access the console server command line, you select SSH as the protocol and use the default IP Port 22.

Click Open and you will be presented with the console server login prompt. (You may also receive a Security Alert that the host's key is not cached, you will need to choose yes to continue.)

Using the Telnet protocol is similarly simple, except you use the default telnet port: port 23.

4.5.3 SSHTERM

Another communications package that may be useful is SSHTerm, an open source package that can be downloaded from http://sourceforge.net/projects/sshtools.

- To use SSHTerm for an SSH terminal session from a Windows client, Select File > New Connection.
- A dialog box appears for your Connection Profile.

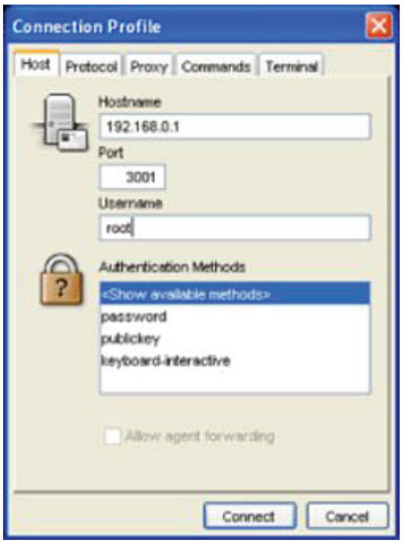


FIGURE 4-17. SSHTERM DIALOG BOX

- Enter the host name or IP address for the console server you are connecting to and the TCP port that the SSH session will use (port 22).
- Enter your username, choose password authentication, and click Connect.
- If you receive a message about the host key fingerprint, select Yes or Always to continue.
- The remote system will prompt you for a username and password. Enter these to login to the console server.

4.6 MANAGEMENT NETWORK CONFIGURATION

The LES1700-R2, LES1516A, LES1532A, LES1548A, LES1508A, and LES1600 console servers have additional network ports that can be configured to provide management LAN access and/or failover or out-of-band access.

4.6.1 ENABLE THE MANAGEMENT LAN

The LES1700-R2, LES1516A, LES1532A, LES1548A, and LES1600 console servers can be configured so the second Ethernet port provides a management LAN gateway. The gateway has firewall, router and DHCP server features. You need to connect an external LAN switch to Network/LAN 2 to attach hosts to this management LAN.

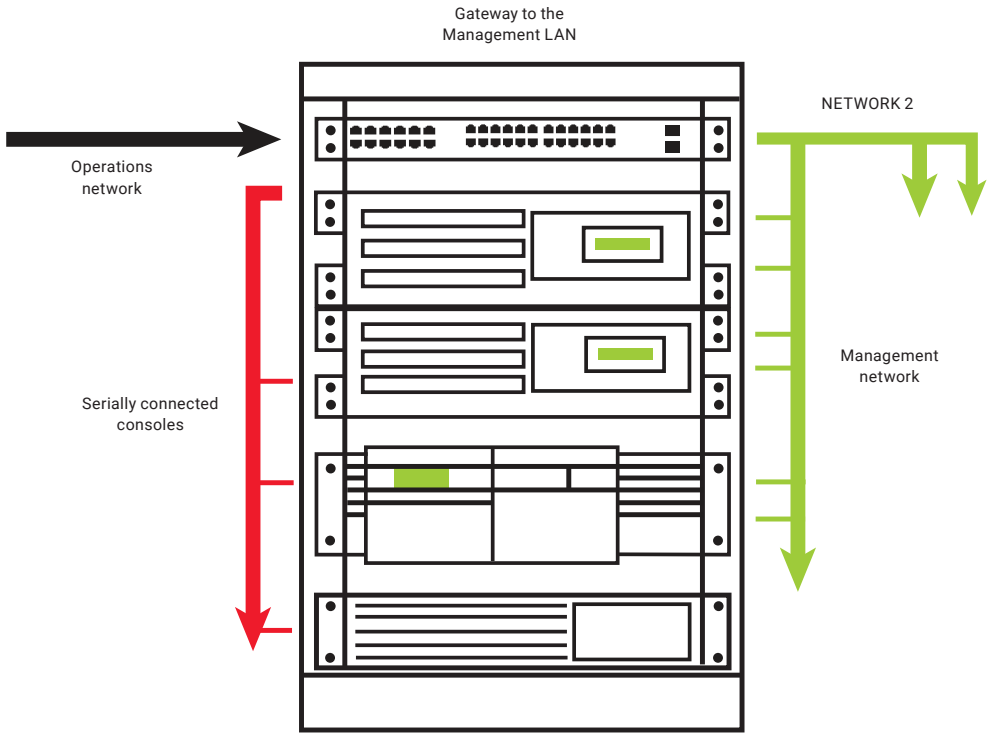


FIGURE 4-18. MANAGEMENT LAN ENABLED

NOTE: The second ethernet port (Network/LAN2) on the LES1700-R2, LES1516A, LES1532A, LES1548A, or LES1600 can be configured as either a Management LAN gateway port or it can be configured as an OOB/Failover port. It cannot be both. Do not allocate Network/LAN 2 as the Failover Interface when you configured the principal Network connection on the System > IP menu.

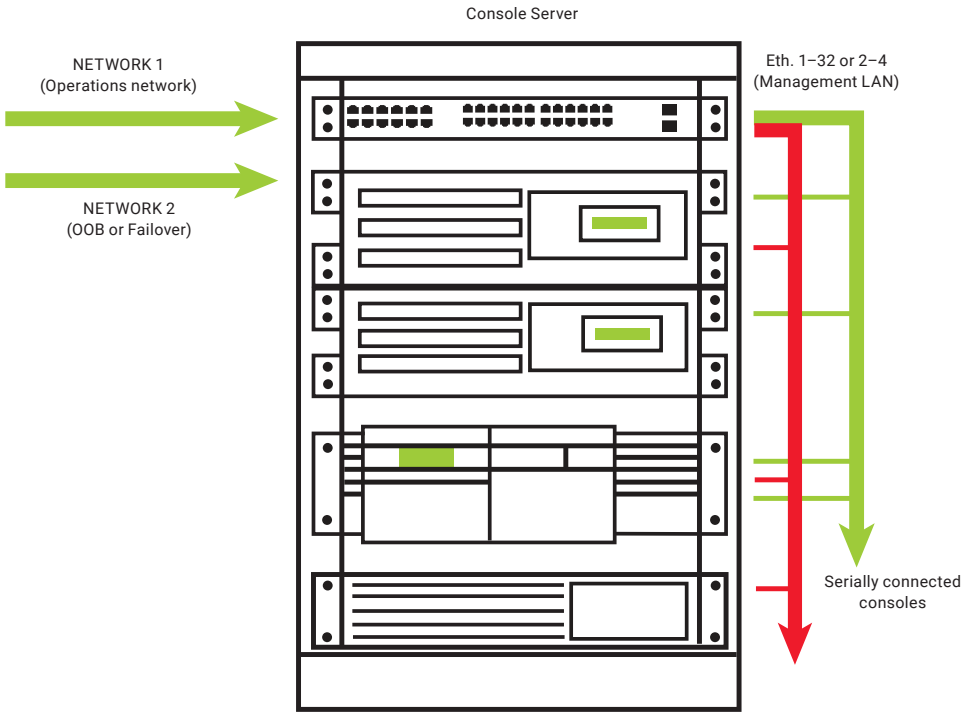


FIGURE 4-19. CONFIGURE AS MANAGEMENT LAN OR OOB/FAILOVER PORT

Management LAN features are disabled by default. To configure a Management LAN gateway:

- Navigate to System > IP.
- Select the Management LAN Interface tab.
- Uncheck Disable.
- Set the IP Address and Subnet Mask for the Management LAN. Leave the DNS fields blank.

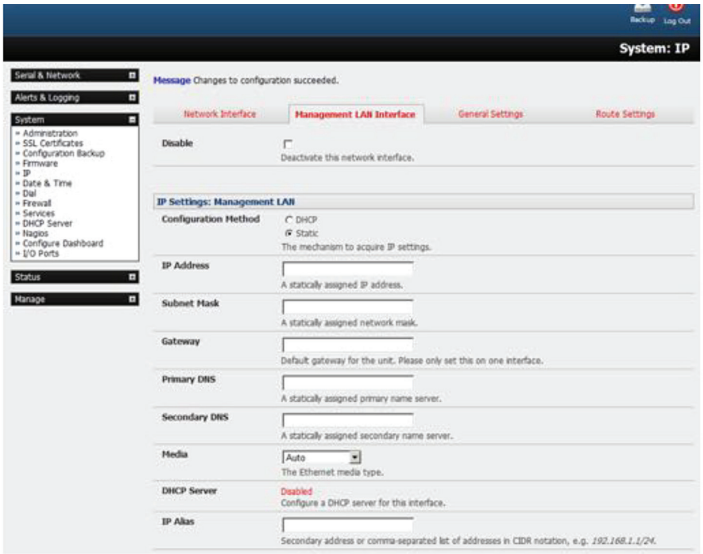


FIGURE 4-20. MANAGEMENT LAN TAB

- Click Apply.

The management gateway function is now enabled with default firewall and router rules. By default, these rules are configured so the Management LAN can only be accessible by SSH port forwarding. This ensures the remote and local connections to Managed Devices on the Management LAN are secure.

The LAN ports can also be configured in bridged or bonded mode (as described later in this chapter) or they can be manually configured from the command line.

4.6.2 CONFIGURE THE DHCP SERVER

All LES1700-R2 and LES1500 family devices host a DHCP server. It is disabled by default. The DHCP server enables the automatic distribution of IP addresses to devices on the Management LAN that are running DHCP clients. To enable the DHCP server:

- Navigate to System > IP.
- Select the Management LAN Interface tab.
- Check the Enable DHCP Server checkbox.
- Enter the Gateway address to be issued to DHCP clients. If left blank, the console server's IP address is used.
- Enter the Primary DNS and Secondary DNS address to be issued to DHCP clients. Again if this field is left blank, the console server's IP address is used. For automatic DNS server assignment, leave this field blank.
- Enter a Domain Name suffix to issue DHCP clients. This is an optional value and step.

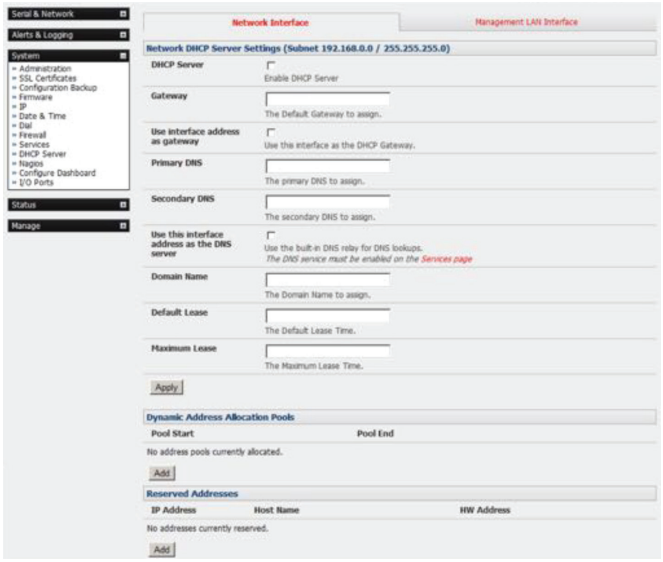


FIGURE 4-21. ENTER DOMAIN NAME SUFFIX (OPTIONAL)

- Enter the Default Lease time and Maximum Lease time in seconds. The lease time is the time that a dynamically assigned IP address is valid before the client must request it again.
- Click Apply.

The DHCP server sequentially issues IP addresses from the specified address pool or pools:

- Click Add in the Dynamic Address Allocation Pools field.

- Enter the DHCP Pool Start Address and End Address.
- Click Apply.

The DHCP server also supports pre-assigning IP addresses to be allocated only to specific MAC addresses and reserving IP addresses to be used by connected hosts with fixed IP addresses. To reserve an IP addresses for a particular host:

- Click Add in the Reserved Addresses field.
- Enter the Hostname, the Hardware Address (MAC) and the Statically Reserved IP address for the DHCP client.
- Click Apply.

When DHCP has initially allocated hosts addresses, we recommend that you copy these into the pre-assigned list so the same IP address will be reallocated in the event of a reboot.

FIGURE 4-22. PRE-ASSIGN IP ADDRESSES

4.6.3 SELECT FAILOVER OR BROADBAND OOB

The LES1700-R2, LES1516A, LES1532A, LES1548A, and LES1600 console servers provide a failover option, so if there is a problem using the main LAN connection for accessing the console server; an alternate access path is used.

By default, the failover is not enabled. To enable:

- Navigate to System > IP.
- Select the Network tab.
- Select the Failover Interface to be used if there is a main network outage. This can be:
  - an alternate broadband Ethernet connection (for example, the Network/LAN2 port on most models) or
  - the LES1700-R2 family internal modem or
  - an external serial modem device connected to the LES1700-R2 Console port (for dialing out to an ISP or the remote management office).

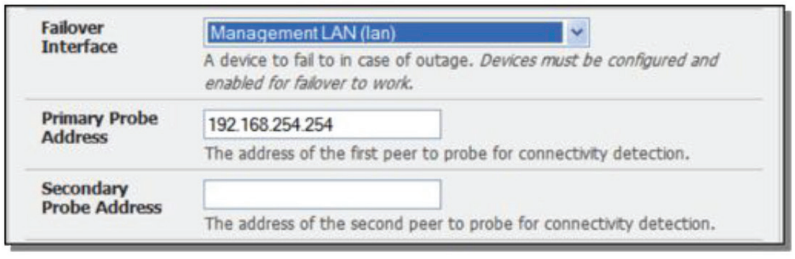


FIGURE 4-23. SELECT FAILOVER INTERFACE

- Click Apply.

NOTE: The failover method is not active until the external sites to be probed to trigger failover are specified and the failover ports themselves are set-up. This is covered in Chapter 6.

NOTE: On the LES1700-R2, LES1516A, LES1532A, LES1548A, and LES1600 models, the second Ethernet port can be configured as either a gateway port or as an OOB/Failover port, but not both.

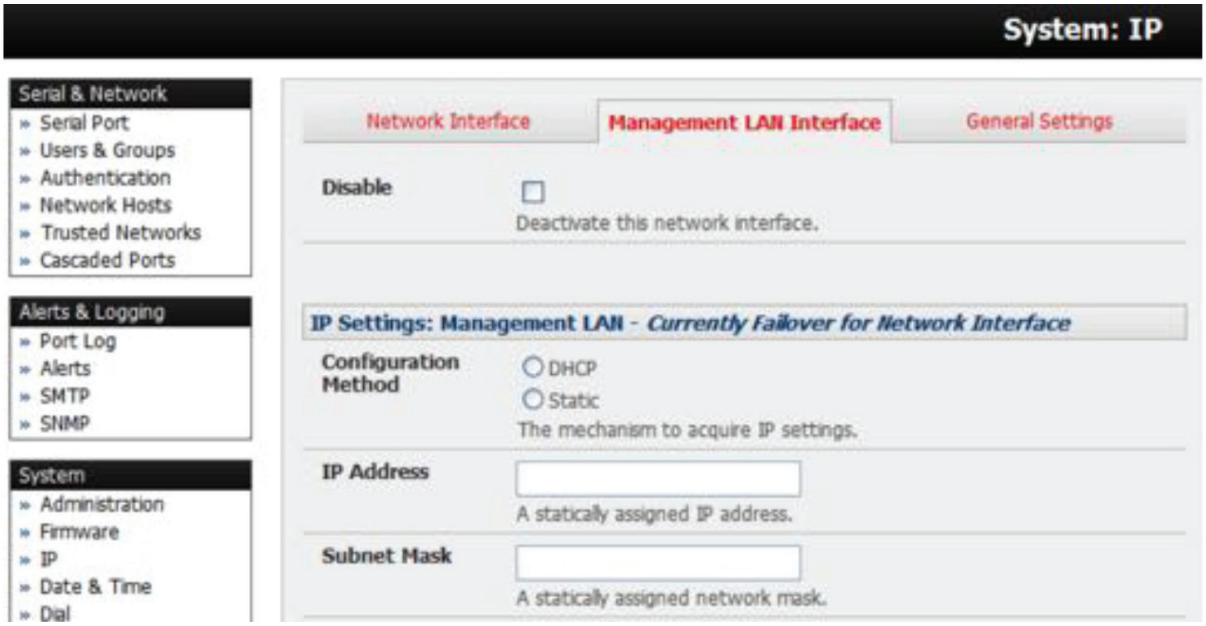


FIGURE 4-24. MANAGEMENT LAN TAB

4.6.4 AGGREGATING THE NETWORK PORTS

By default, the console server's Management LAN network ports can only be accessed using SSH tunneling/port forwarding or by establishing an IPsec VPN tunnel to the console server.

All the wired network ports on the console servers can be aggregated by being bridged or bonded.

- Navigate to System > IP.
- Click the General Settings tab.
- Click the Bridge Interfaces or Bond Interfaces radio button to enable wired Ethernet interface aggregation.

When bridging is enabled, network traffic is forwarded across all Ethernet ports with no firewall restrictions. All Ethernet ports are transparently connected at the data link layer (layer 2) so they do retain their unique MAC addresses.

With bonding, the network traffic is carried between the ports but they present with one MAC address.

Both modes remove all the Management LAN Interface and Out-of-Band/Failover Interface functions and disable the DHCP Server.

NOTE: In aggregation mode, all the Ethernet ports are configured collectively via the System > IP > Network Interface tab.

4.6.5 STATIC ROUTES

Firmware 3.4 and later support static routes that provide a quick way to route data from one subnet to different subnet. You can hard-code a path that specifies the console server or router to get to a certain subnet by using a certain path. This may be useful for remotely accessing various subnets at a remote site when being accessed using the cellular OOB connection.

To add a static route to the route table of the system:

- Navigate to System > IP > General Settings.
- Select the Route Settings tab.

FIGURE 4-25. ROUTE SETTINGS SCREEN

- Enter a meaningful Route Name for the route.
- In the Destination Network/Host field, enter the IP address of the destination network or host that the route provides access to.
- Enter a value in the Destination netmask field that identifies the destination network or host. Any number between 0 and 32. A subnet mask of 32 identifies a host route.
- Fill the Route Gateway field with the IP address of a router that will route packets to the destination network. This field may be left blank, depending on your network configuration.
- Select the Interface to use to reach the destination This field may be left as None.
- Enter a value in the Metric field that represents the metric of this connection. This generally only has to be set if two or more routes conflict or have overlapping targets. Any number equal to or greater than 0.
- Click Apply.

NOTE: The route details page provides a list of network interfaces and modems to which a route can be bound. In the case of a modem, the route will be attached to any dialup session that is established via that device. A route can be specified with a gateway, an interface or both. If the specified interface is not active for whatever reason, then routes configured for that interface will not be active.

4.7 CONFIGURATION OVER DHCP (ZTP)

Config-over-DHCP is available for all Black Box console managers running firmware release 3.16 or later. Using this feature, Black Box devices can be provisioned during their initial boot from a DHCP server. Provisioning on untrusted networks can be facilitated by providing keys on a USB flash drive.

The typical steps for configuration over a trusted network are:

- Manually configure a same-model Black Box device.
- Save its configuration as an Black Box backup (.opg) file.
- Select System > Configuration Backup > Remote Backup.
- Click Save Backup.

A backup configuration file — model-name_iso-format-date_config.opg — is downloaded from the Black Box device to the local system.

Alternatively, you can save the configuration as an xml file:

- Select System > Configuration Backup > XML Configuration. An editable field containing the configuration file in XML format is presented.
- Click into the field to make it active.
- If you are running any browser on Windows or Linux, right-click and choose Select All from the contextual menu or press Control-A. Then right-click and choose Copy from the contextual menu or press Control-C.
- If you are using any browser on macOS, choose Edit > Select All or press Command-A. Then choose Edit > Copy or press Command-C.
- In your preferred text-editor, create a new empty document, paste the copied data into the empty document and save the file. Whatever file-name you choose, it must include the.xml filename suffix.
- Copy the saved .opg or .xml file to a public-facing directory on a file server serving at least one of the following protocols: HTTPS, HTTP, FTP or TFTP.

NOTE: Only HTTPS can be used if the connection between the file server and a to-be-configured Black Box device travels over an untrusted network.

- Configure your DHCP server to include a vendor specific option for Black Box devices. (This will be done in a DHCP server-specific way.) The vendor specific option should be set to a string containing the URL of the published .opg or .xml file in the step above. The option string must not exceed 250 characters and it must end in either .opg or .xml.
- Connect a new Black Box device (either factory-reset or Config-Erased) to the network and apply power.

NOTE: It may take up to 5 minutes for the device to find the .opg or .xml file via DHCP, download and install the file, and then reboot itself.

4.7.1 ENSURING THE CONSOLE SERVER IS UNCONFIGURED

Console servers exist in two states: configured or unconfigured. For ZTP via Config-over-DHCP to work, a target console server must be in an unconfigured state.

Console servers ship unconfigured from the factory: assuming a compatible configuration file is to hand (see below), a newly-unboxed console server can be configured using ZTP.

To return an already-configured console server to its unconfigured state do ether of the following:

- While the console server is powered-on, press the recessed Erase button twice. This button is found on the rear or side of every console server.

Alternatively:

- Navigate to System > Administration.
- Check the Config Erase checkbox.
- Check the Reboot checkbox.
- Click Apply.

NOTE: If ZTP is being used to update a working console server's firmware, the extant configuration must be backed-up before the console server is unconfigured.

4.7.2 EXAMPLE ISC DHCP (DHCPD) SERVER CONFIGURATION

The following is an example DHCP server configuration fragment for serving an .opg configuration image via the ISC DHCP server, dhcpd:

```
option space Black Box  code width 1  length width 1;
option Black Box.config-url code 1 = text;
class "Black Box-config-over-dhcp-test" {
   match if option vendor-class-identifier ~~ "^Black Box/";
   vendor-option-space Black Box;
   option Black Box.config-url
   "https://example.com/opg/${class}.opg";
}
```

4.7.3 SETUP WHEN THE LAN IS UNTRUSTED

If the connection between the file server and a to-be-configured Black Box device includes an untrusted network, a two-handed approach can mitigate the issue.

NOTE: This approach introduces two physical steps where trust can be difficult, if not impossible, to establish completely. First, the custody chain from the creation of the data-carrying USB flash drive to its deployment. Second, the hands connecting the USB flash drive to the Black Box device.

- Generate an X.509 certificate for the Black Box device.
- Concatenate the certificate and its private key into a single file named client.pem.
- Copy client.pem onto a USB flash drive.
- Set up an HTTPS server so that access to the .opg or .xml file is restricted to clients that can provide the X.509 client certificate generated above.
- Put a copy of the CA cert that signed the HTTP server's certificate—ca-bundle.crt—onto the USB flash drive bearing client.pem.
- Insert the USB flash drive into the Black Box device before attaching power or network.
- Continue the procedure from 'Copy the saved .opg or .xml file to a public-facing directory on a file server' above using the HTTPS protocol between the client and server.

4.7.4 PREPARE A USB DRIVE AND CREATE THE X.509 CERTIFICATE AND PRIVATE KEY

- Generate the CA certificate so the client and server Certificate Signing Requests (CSRs) can be signed.

```
# cp /etc/ssl/openssl.cnf .
# mkdir -p exampleCA/newcerts
# echo 00 > exampleCA/serial
# echo 00 > exampleCA/crlnumber
# touch exampleCA/index.txt
# openssl genrsa -out ca.key 8192
# openssl req -new -x509 -days 3650 -key ca.key -out demoCA/
 \cacert.pem -subj /CN=ExampleCA
# cp demoCA/cacert.pem ca-bundle.crt
```

NOTE: This procedure generates a certificate called ExampleCA but any allowed certificate name can be used. Also, this procedure uses openssl ca. If your organization has an enterprise-wide, secure CA generation process, that should be used instead.

- Generate the client certificate.

```
# openssl genrsa -out client.key 4096
# openssl req -new -key client.key -out client.csr -subj \
 /CN=ExampleClient
# openssl ca -days 365 -in client.csr -out client.crt \
      -keyfile ca.key -policy policy_anything -batch -notext
# cat client.key client.crt > client.pem
```

- Format a USB flash drive as a single FAT32 volume.
- Move the client.pem and ca-bundle.crt files onto the flash drive's root directory.

4.7.5 WHAT AN UNCONFIGURED CONSOLE SERVER DOES ON FIRST BOOT

When an unconfigured console server boots the following steps occur:

- the console server starts the udhcpc process (via conman).
- udhcpc transmits a DHCP DISCOVER request to the primary Network Interface.

This request includes a Vendor Class Indentifier in the following form:

Black Box/model-name

For example:

Black Box/LES1203A-M

NOTE: In unconfigured console servers, the network interface mode is unset and the DHCP DISCOVER request, therefore, includes a parameter request for Vendor-Specific Information (option 43). Configured console servers have a config.interfaces.wan. mode with configuration information included. Consequently, the DHCP DISCOVER packet sent from such servers does not include an option 43 request.

- the DHCP server sends a DHCP OFFER in reply.

The console server uses the information in the DHCP OFFER to

- assign itself the supplied IPv4 address.
- add a default route.
- prepare its DNS resolver.

If the DHCP OFFER also includes an option 43 with sub-option 1, the console server:

- reads the contents of sub-option 1 as a white-space delimited list of URLs.
- interprets the URLs as locations for configuration files to use as to configure itself.
- temporarily stores the URLs for later use.

If the DHCP OFFER also includes an option 43 with sub-option 2, the console server:

- reads the contents of sub-option 2 as a white-space delimited list of URLs.
- interprets the URLs as locations for firmware images to use to flash the firmware on itself.
- temporarily stores the URLs for later use.

If the DHCP OFFER also includes a URL to an NTP server, the console server:

- syncronizes its system clock to the referenced NTP server.

see etc/scripts/udhcpc.script for details.

4.7.6 USING UNCONFIGURED CONSOLE SERVER ON FIRST BOOT TO UPDATE FIRMWARE

This process requires three things:

- a console server running firmware 3.16.6 or later.
- a file containing the current configuration of the console server to be updated available at a working URL that is declared in option 43, sub-option 1 of your DHCP server's DHCP OFFER.
- the firmware image to be applied available at a working URL that is declared in option 43, sub-option 2 of your DHCP server's DHCP OFFER.

The working URLs can be offered over ftp, tftp, http, and https. However, for https to work, the console server must be in secure recovery mode. See Section 4.7.9 for secure recovery mode requirements.

When the console server having its firmware updated is unconfigured and restarted, it:

- runs /etc/scripts/backup-url\ loadimage for each URL included in option 43 sub-option 2 of the DHCP OFFER.

On the first URL to return a firmware image, the console server:

- runs curl to download the firmware image.
- passes the image to netflash as standard input.

netflash then:

- checksums and flashes the passed-in image.
- reboots the console server.

NOTE: netflash will not reboot the console server unless the image passes the checksum.

Upon rebooting, the console server:

- runs etc/config/.init to process the firmware image.

- runs etc/scripts/backup-url to restore the backed-up configuration using the file declared in option 43, sub-option 1 of the DHCP OFFER. (The script's name is historical: it is based on configuration backup and restore logic.)

4.7.7 THE URLS IN DCHP OFFER, OPTION 43, SUB-OPTION 1

URLs offered in DCHP OFFER, option 43, sub-option 1 are parsed by /etc/scripts/backup-url using substrings in the configuration backup's filename to determine the choice order. The order is as follows.

TABLE 4-2. CHOICE ORDER FOR URLS

SUB-STRING	REPLACED BY	EXAMPLE
${mac}	the device's 12-digit MAC address, in lowercase	0013b600b669
${model}	the device's full model name, in lowercase	les1708a-r2
${class}	the firmware's hardware class	les1700-r2
${version}	the firmware's version number	4.1.0u3

Once downloaded, a configuration file is checked:

- if it is a .opg file, its header is checked for compatibility with the current device.
- if it is a .xml file, a parse check is made.

In both cases, if the check fails, the downloaded file is abandoned and the next URL is tried.

4.7.8 IMPORTING THE CONFIGURATION FILE

Once a downloaded configuration file passes the appropriate check, the console server:

- imports the downloaded and checked configuration file.
- checks the configuration file for a hostname to set itself to.

If no hostname can be set, the console server defaults to

${model}-${mac}

(That is, it sets its hostname to the device's full model name, followed by a hyphen, followed by the device's MAC address.)

- checks that it is still unconfigured.
- sets the network interface mode to DHCP.

This, in effect, forces the console server into a configured state, preventing a reboot loop from occurring.

- returns a reboot-necessary flag.

This last action ensures the now configured console server reboots.

4.7.9 RUNNING A RESTORE OR UPDATE IN SECURE RECOVERY MODE

For a firmware update to run in secure mode (that is, to run over the https protocol) /etc/scripts/backup-url must find two certificate files in an attached USB storage device.

The first required file is ca-bundle.crt. The second required file is whichever one of the following files is found first:

- client-AABBCCDDEEFF.pem

AABBCCDDEEFF is the MAC address of the console server's primary network interface.

- client-MODEL.pem

MODEL is the (vendor class) model name in lowercase, truncated to before the first hyphen.

- lient.pem

See Section 4.7.4 for how to create these files.

NOTE: If both ca-bundle.crt and a suitable *.pem file are found, URLs offered by insecure protocols (such as http, ftp, tftp and ftps) are skipped. Once an unconfigured console server is in secure recovery mode, the firmware and configuration files needed to return it to operational status must be offered via https.

CHAPTER 5: SERIAL PORT, HOST DEVICE AND USER CONFIG

The console server enables access and control of serially-attached devices and network-attached devices (hosts). The Administrator must configure access privileges for each of these devices, and specify the services that can be used to control the devices. The Administrator can also set up new users and specify each user's individual access and control privileges.

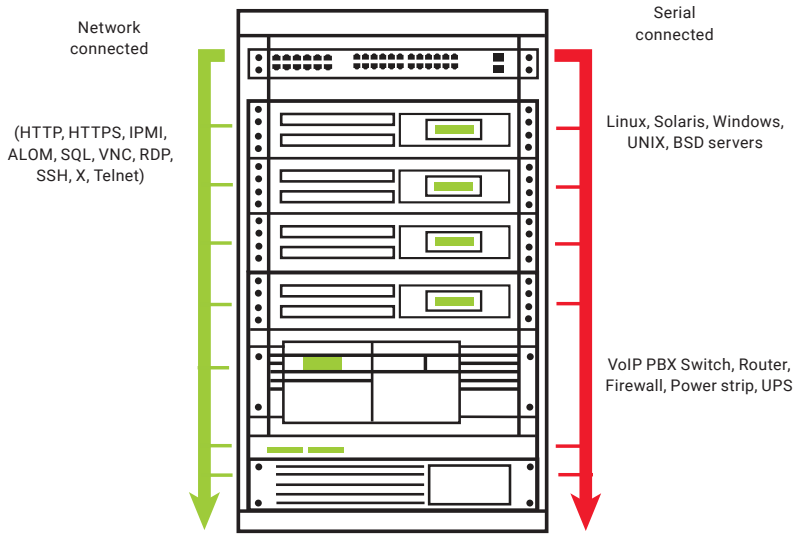


FIGURE 5-1.

This chapter covers each of the steps in configuring network-connected and serially-attached devices.

TABLE 5-1. STEPS SUMMARY

STEP	NOTES
Serial ports	Setting up serially connected device protocols
Users & Groups	Setting up and defining user access permissions
Authentication	Also covered in more detail in Chapter 10
Network hosts	Configuring access to network-connected hosts
Configuring trusted networks	Nominate IP addresses trusted users access from
Serial console port cascading and redirection	—
Power (UPS, PDU and IPMI)	—
Environmental Monitoring Devices (EMD)	—
Serial port redirection	The PortShare client on Windows and Linux
Managed devices	The consolidated view of all the connections
IPSec	Enabling VPN connections
OpenVPN	—
PPTP	—

5.1 CONFIGURE SERIAL PORTS

The first step in configuring a serial port is to set the Common Settings such as the protocols and the RS-232 parameters that are to be used for the data connection to that port (for example, baud rate).

Then you select what mode the port is to operate in. Each port can be set to support one of the operating modes in the next table.

TABLE 5-2. OPERATING MODES

MODE	NOTES
Disabled	The serial port is inactive
Console server	Enables general access to serial console port on the serially attached devices
Device	Sets the serial port up to communicate with an intelligent serial controlled PDU, UPS or Environmental Monitor Devices (EMD)
SDT	Enables graphical console access (with RDP, VNC, HTTPS etc.) to hosts that are serially connected
Terminal server	Sets the serial port to await an incoming terminal login session
Serial bridge	Enables the transparent interconnection of two serial port devices over a network

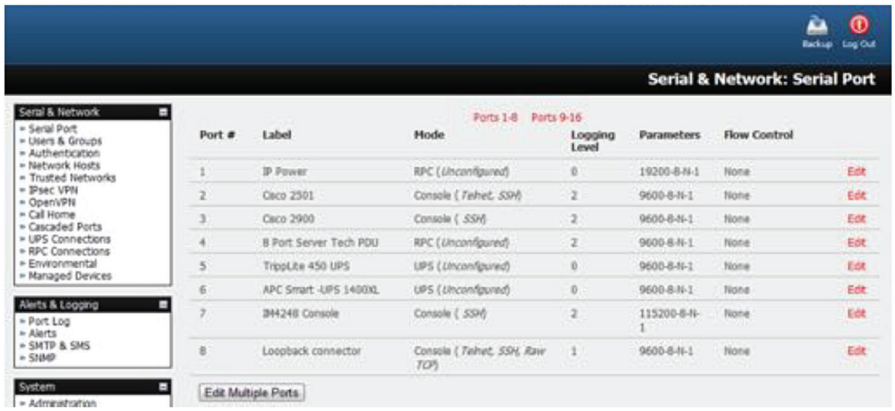


FIGURE 5-2. SERIAL & NETWORK: SERIAL PORT SCREEN

- Navigate to Serial & Network > Serial Port. Details of the currently setup serial ports presents. By default, each serial port is set in console server mode.
- Click Edit to reconfigure a given serial port.
- Reconfigure the common settings (Section 5.1.1) and the mode Sections 5.1.2–5.1.6) for each port as needed.
- Set up any remote syslog (Section 5.1.7).
- Click Apply.

NOTE: To set the same protocol options for multiple serial ports at once click Edit Multiple Ports and select which ports you wish to configure as a group.

- If the console server has been configured with distributed Nagios monitoring enabled, then you will also be presented with Nagios Settings options to enable nominated services on the host to be monitored (see Chapter 11).

5.1.1 COMMON SETTINGS

There are a number of common settings that can be set for each serial port. These are independent of the mode in which the port is being used. These serial port parameters must be set so they match the serial port parameters on the device you attach to that port.

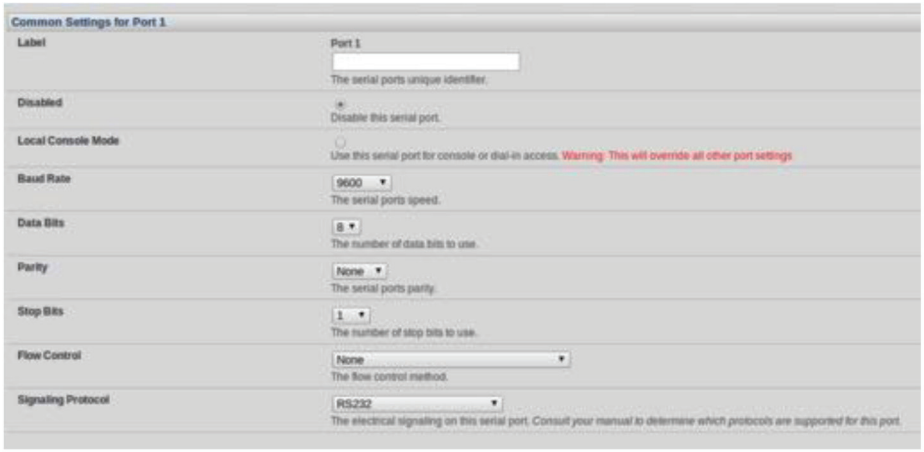


FIGURE 5-3.

- Specify a Label for the port.
- Select the appropriate Baud Rate, Parity, Data Bits, Stop Bits and Flow Control for each port.
- Set the Signaling Protocol. This menu item only presents in ports with RS-422/485 options (all ports on LES1204A-2, LES1508A). The options available are RS-232, RS-422, RS-485 and RS-485 Echo mode.
- Set the Port Pinout. This menu item only presents for LES1700-R2 ports where pinout for each RJ-45 serial port can be set as either X2 (Cisco Straight) or X1 (Cisco Rolled).
- Before proceeding with further serial port configuration, you should connect the ports to the serial devices they will be controlling, and ensure they have matching settings.

5.1.2 CONSOLE SERVER MODE

- Select Console Server Mode to enable remote management access to the serial console that is attached to this serial port.

FIGURE 5-4. CONSOLE SERVER SETUP SCREEN

- Set the desired Logging Level. This specifies the level of information to be logged and monitored (see Chapter 8).
- Enable or disable Telnet access.

When the Telnet service is enabled on the console server, a Telnet client on a User's or Administrator's computer can connect to a serial device attached to this serial port on the console server. Telnet communications are unencrypted so this protocol is generally recommended only for local or VPN-tunneled connections.

Windows 2000, Windows XP and Windows NT can run telnet from the cmd.exe command prompt.

Windows Vista and later ship with a Telnet client but it is not enabled by default. You can install it as follows.

- Click the Start button.
- Click Control Panel.
- Click Programs.
- Click Turn Windows features on or off.

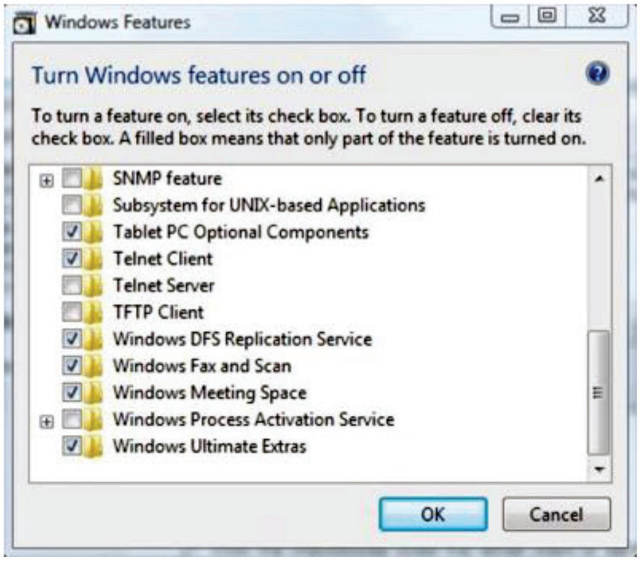


FIGURE 5-5. TURN WINDOWS FEATURES ON OR OFF

If you are prompted for an administrator password or confirmation, type the password or provide confirmation.

- In the Windows Features dialog box, select the Telnet Client check box.
- Click OK.

The installation may take several minutes.

If remote communications are being tunneled with SDT Connector, then Telnet can be used for securely accessing these attached devices.

NOTE: In Console Server mode, Users and Administrators can use SDT Connector to set up secure Telnet connections that are SSH tunneled from their client computers to the serial port on the console server. SDT Connector can be installed on Windows PCs and on most Linux platforms and it enables secure Telnet connections to be selected with a simple point-and-click. To use SDT Connector to access consoles on the console server serial ports, you configure SDT Connector with the console server as a gateway, then as a host, and you enable Telnet service on Port 2000 + serial port # (that is Ports 2001–2048). See Chapter 7 for more details on using SDT Connector for Telnet and SSH access to devices that are attached to the console server serial ports.

You can also use communications packages like PuTTY to set a direct Telnet (or SSH) connection to the serial ports.

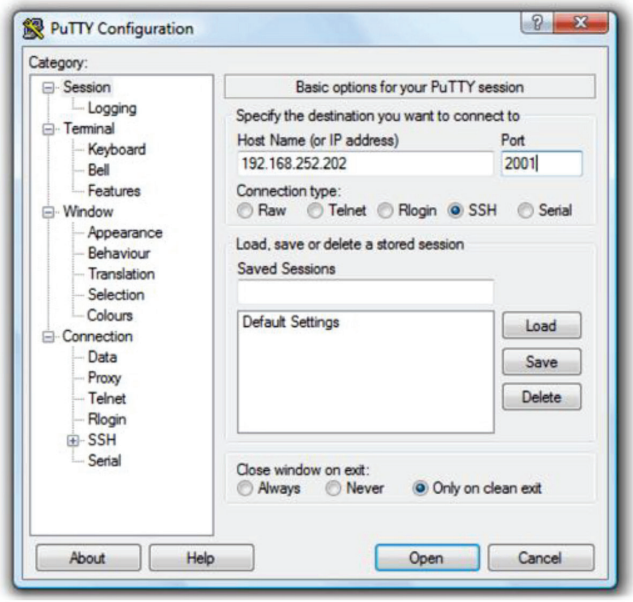


FIGURE 5-6. PUTTY CONFIGURATION SCREEN

NOTE: PuTTY supports Telnet (and SSH). Enter the console server's IP address as the Host Name (or IP address). Select Telnet as the protocol and set the TCP port to 2000 plus the physical serial port number (that is a port between 2001 and 2048). Click the Open button. You may receive a Security Alert that the host's key is not cached: choose yes to continue. The login prompt of the remote system connected to the serial port chosen on the console server will now present. You can login as normal and use the host serial console screen.

Putty can be downloaded from http://putty.org/.

NOTE: In Console Server mode, when you connect to a serial port you connect via pmshell. To generate a BREAK on the serial port type the character sequence ~b. If you're doing this over OpenSSH type ~~b.

- Enable or disable SSH access.

  We recommend that you use SSH as the protocol where the User or Administrator connects to the console server (or connects through the console server to the attached serial consoles) over the Internet or any other public network. This will provide authenticated SSH communications between the SSH client program on the remote user's computer and the console server, so the user's communication with the serial device attached to the console server is secure.

  For SSH access to the consoles on devices attached to the console server serial ports, you can use SDT Connector. You configure SDT Connector with the console server as a gateway, then as a host, and you enable SSH service on Port 3000 + serial port #. (That is ports 3001 – 3048). See Chapter 7 for more information on using SDT Connector for SSH access to devices that are attached to the console server serial ports.

  Also you can use common communications packages, like PuTTY or SSHTerm to SSH connect directly to port address IP Address:Port 3000 + serial port #. (That is ports 3001 – 3048).

  Alternately, SSH connections can be configured using the standard SSH port 22. The serial port being accessed is then identified by appending a descriptor to the username. This syntax supports any of the following descriptors:

  <username>:<portXX>

  <username>:<port-label>

  <username>:<ttySX>

  <username>:<serial>

For example, if a User named fred wants to access serial port 2, when setting up SSHTerm or the PuTTY SSH client, instead of typing

username = fred

ssh port = 3002

type

username = fred:port02

or

username = fred:ttyS1

and

ssh port = 22.

Alternatively, by typing

username=fred:serial

and

ssh port = 22

the User is presented with a port selection option

This syntax enables Users to set up SSH tunnels to all serial ports with only a single IP port 22 having to be opened in their firewall or gateway.

NOTE: In Console Server mode, when you connect to a serial port, you connect via pmshell. To generate a BREAK on the serial port type the character sequence ~b. If you're doing this over OpenSSH, type ~~b.

- Enable or disable Raw TCP access.

  RAW TCP allows connections directly to a TCP socket. Communications programs like PuTTY support RAW TCP. This protocol, however, would usually be used by a custom application.

  For RAW TCP, the default port address is IP Address:Port 4000 + serial port # (That is, ports 4001 – 4048).

  RAW TCP also enables the serial port to be tunneled to a remote console server, so two serial port devices can be transparently interconnect over a network (see Section 5.1.6).

- Enable or disable RFC 2217 access.

  Enabling RFC 2217 access enables serial port redirection on that port. For RFC 2217, the default port address is IP Address:Port 5000 + serial port # (that is Port #s 5001 – 5048).

  Special client software is available for Windows UNIX and Linux that supports RFC 2217 virtual com ports, so a remote host can monitor and manage remote serially attached devices, as though they were connected to the local serial port (see Section 5.6 for details).

  RFC 2217 also enables the serial port to be tunneled to a remote console server, so two serial port devices can be transparently interconnect over a network (see Section 5.1.6).

- Enable or disable Unauthenticated Telnet.

  Enabling Unauthenticated Telnet enables telnet access to the serial port without authentication credentials. When a user accesses the console server to telnet to a serial port, they are normally given a login prompt. With unauthenticated telnet, they connect directly through to the port without any console server login challenge. (If a telnet client does prompt for authentication, any entered data will allow connection.)

  This mode is mainly used when you have an external system (such as conserver) managing user authentication and access privileges at the serial device level.

NOTE: Only the connection to the console server is unauthenticated. Logging into a device connected to the console server may still require authentication.

For Unauthenticated Telnet the default port address is IP Address:Port 6000 + serial port # (that is Port #s 6001 – 6048).

- Enable or disable Web Terminal.

  Enabling Web Terminal enables web browser access to the serial port via Manage > Devices > Serial using the Management Console's built in AJAX terminal.

  Web Terminal connects as the currently authenticated Management Console user and does not re-authenticate. See Section 14.3 for more details.

- Enter an IP Alias (for the Network Interface, Management LAN or Out-of-Band/Failover).

  A working IP Alias enables access to the serial port using a specific IP address, specified in CIDR format. Each serial port can be assigned one or more IP aliases, configured on a per-network-interface basis.

  A serial port can, for example, be made accessible at both 192.168.0.148 (as part of the internal network) and 10.10.10.148 (as part of the Management LAN). It is also possible to make a serial port available on two IP addresses on the same network (for example, 192.168.0.148 and 192.168.0.248).

  These IP addresses can only be used to access the specific serial port, accessible using the standard protocol TCP port numbers of the console server services. For example, SSH on serial port 3 would be accessible on port 22 of a serial port IP alias (whereas on the console server's primary address, it is available on port 2003).

  This feature can also be configured via the multiple port edit page. In this case, the IP addresses are applied sequentially, with the first selected port getting the IP entered and subsequent ones getting incremented, with numbers being skipped for any unselected ports. For example if ports 2, 3 and 5 are selected and the IP alias 10.0.0.1/24 is entered for the Network Interface, the following addresses will be assigned:

  Port 2: 10.0.0.1/24

  Port 3: 10.0.0.2/24

  Port 5: 10.0.0.4/24

- Enable or disable Encrypt Traffic and enable or disable Authenticate. (These options should be either enabled or disabled as a pair.)

  Enabling these two options turns on trivial encryption and authentication of RFC2217 serial communications using Portshare. For strong encryption, use VPN.

- Set an Accumulation Period.

  Once a connection has been established for a particular serial port (such as a RFC2217 redirection or Telnet connection to a remote computer) any incoming characters on that port are forwarded over the network on a character by character basis. The accumulation period changes this by specifying a period of time that incoming characters will be collected before then being sent as a packet over the network.

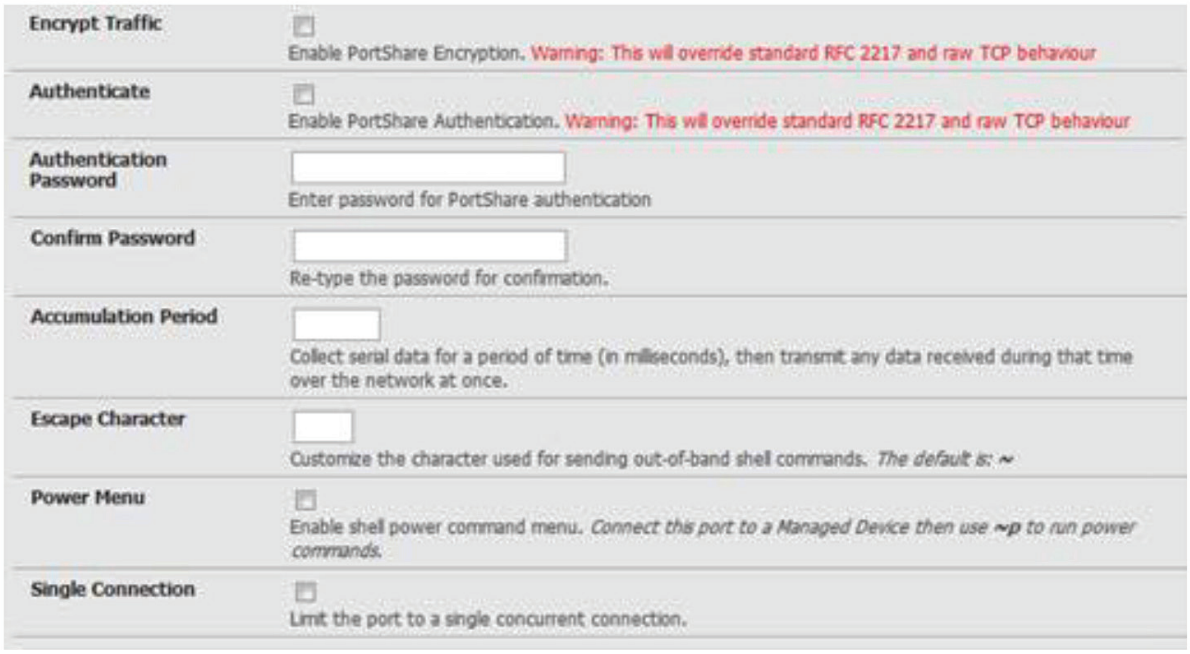


FIGURE 5-7.

- Set a custom Escape Character. This enables you to change the character used for sending escape characters. The default is ~.
- Enable or disable the Power Menu.

5.1.3 SDT MODE

This Secure Tunneling setting allows port forwarding of RDP, VNC, HTPP, HTTPS, SSH, Telnet and other LAN protocols through to computers which are locally connected to the console server by their serial COM port. However such port forwarding requires a PPP link to be set up over this serial port.

FIGURE 5-8. SDT SETTINGS SCREEN

For configuration details, refer to Section 7.6.

5.1.4 DEVICE (RPC, UPS, EMD) MODE

This mode configures the selected serial port to communicate with a serial controlled Uninterruptable Power Supply (UPS), Remote Power Controller / Power Distribution Units (RPC) or Environmental Monitoring Device (EMD).

FIGURE 5-9. DEVICE SETTINGS SCREEN

- Select the desired Device Type (UPS, RPC or EMD).
- Proceed to the appropriate device configuration page: Serial & Network > UPS Connections, RPC Connection or Environmental) as detailed in Chapter 9.

5.1.5 TERMINAL SERVER MODE

- Enable Terminal Server Mode and set the Terminal Type (vt220, vt102, vt100, Linux or ANSI) to enable a getty on the selected serial port.

FIGURE 5-10. TERMINAL SERVER SETTINGS SCREEN

The getty will then configure the port and wait for a connection to be made. An active connection on a serial device is usually indicated by the Data Carrier Detect (DCD) pin on the serial device being raised. When a connection is detected, the getty program issues a login: prompt, and then invokes the login program to handle the actual system login.

NOTE Selecting Terminal Server mode will disable Port Manager for that serial port, so data is no longer logged for alerts etc.

5.1.6 SERIAL BRIDGING MODE

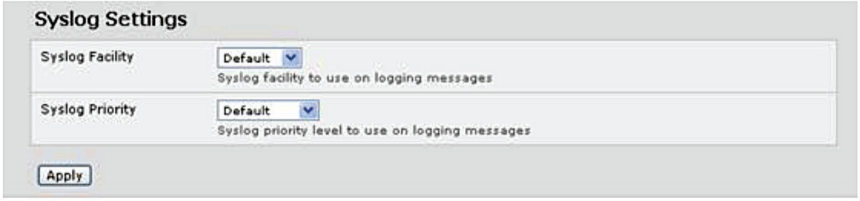


FIGURE 5-11. TERMINAL SERVER SETTINGS SCREEN

With serial bridging, the serial data on a nominated serial port on one console server is encapsulated into network packets and then transported over a network to a second console server where is then represented as serial data. So the two console servers effectively act as a virtual serial cable over an IP network.

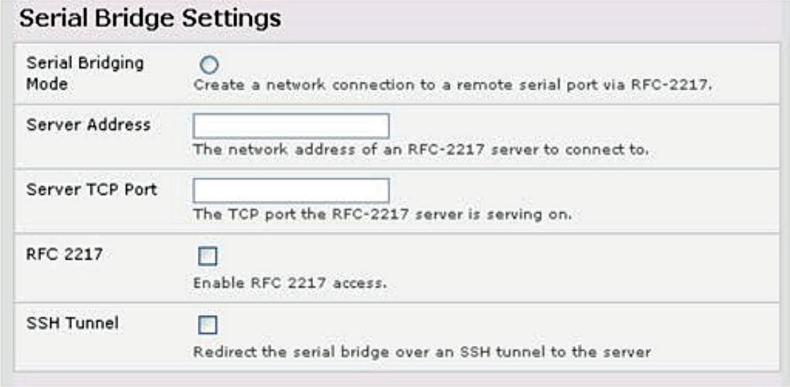


FIGURE 5-12.

One console server is configured to be the Server. The Server serial port to be bridged is set in Console Server mode with either RFC2217 or RAW enabled (as described in Section 5.1.2).

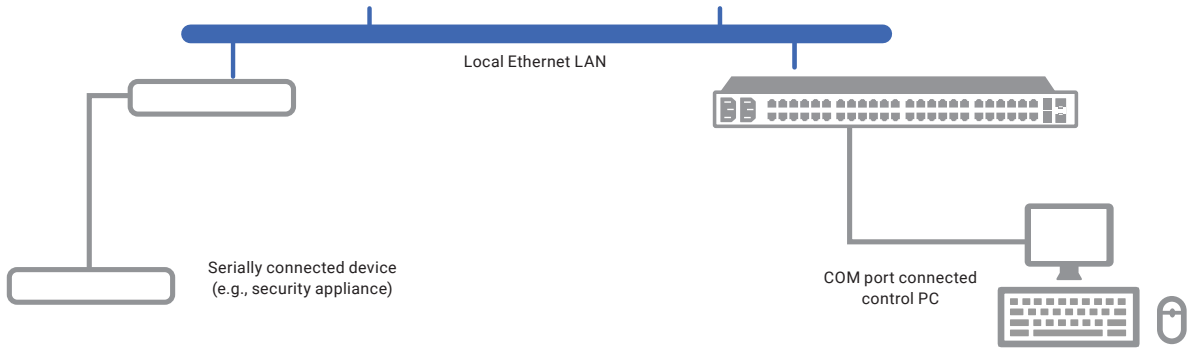


FIGURE 5-13.

For the Client console server, the serial port to be bridged must be set in Bridging Mode.

- Enable Serial Bridging Mode and specify the IP address of the Server console server and the TCP port address of the remote serial port (for RFC2217 bridging this will be 5001-5048).
- By default the bridging client will use RAW TCP so you must select RFC2217 if this is the console Server mode you have specified on the server console server.
- You may secure the communications over the local Ethernet by enabling SSH however you will need to generate and upload keys (see Chapter 16).

5.1.7 SYSLOG

In addition to built-in logging and monitoring (which can be applied to serial-attached and network-attached management accesses, as covered in Chapter 8), the console server can also be configured to support the remote syslog protocol on a per serial port basis.

- Select the Syslog Facility and Syslog Priority fields to enable logging of traffic on the selected serial port to a syslog server and to appropriately sort and action those logged messages (for example, redirect them or send an alert email).

For example, if the computer attached to serial port 3 should never send anything out on its serial console port, the Administrator can set the Syslog Facility for that port to local0 (local0 – local7 are meant for site local values), and the Syslog Priority to critical. At this priority, if the console server syslog server does receive a message, it will automatically raise an alert. See Chapter 8 for more.

5.1.8 NMEA STREAMING

The LES1600 can provide GPS NMEA data streaming from the internal GPS /cellular modem. This data stream presents as a serial data stream on port 5.

The Common Settings (baud rate, etc.) are ignored when configuring the NMEA serial port. You can specify the Fix Frequency (i.e. this GPS fix rate determines how often GPS fixes are obtained). You can also apply all the Console Server Mode, Syslog and Serial Bridging settings to this port.

NOTE: The NMEA Streaming menu item should display on the Serial & Network > Serial Port menu.

setfset -r lists all of the current feature set variables.

You look for the factory_opts variable, and then add 3g-gps to it.

For example, factory_opts=rs485,3g,ind.

To update it to 3g-gps, you do the following:

setfset -u factory_opts=rs485,3g-gps,ind.

Then run setfset -r again, and make sure you can see the update.

You can use pmshell, webshell, SSH, RFC2217 or RawTCP to get at the stream.

FIGURE 5-14.

For example, using the Web Terminal:

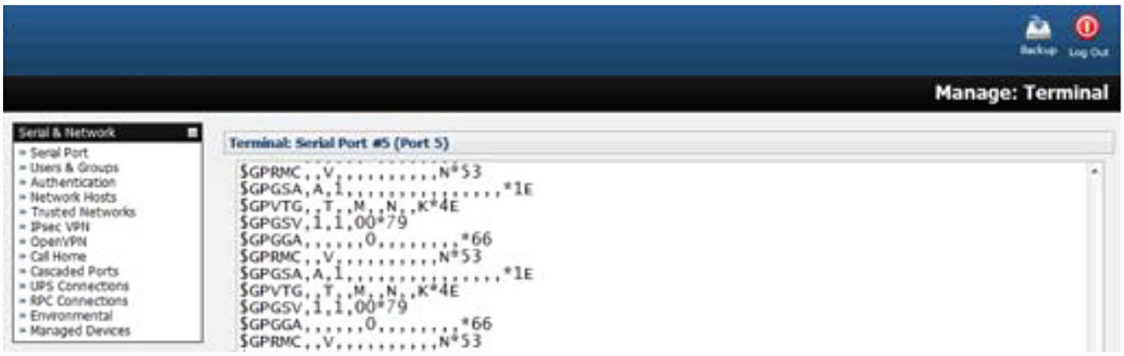


FIGURE 5-15. MANAGE TERMINAL SCREEN

5.1.9 USB PORTS

Black Box LES1600, LES1516A, LES1532A, LES1548A and LES1700-R2 family console servers running firmware 3.16.5 or later support USB console connections to devices from a wide range of vendors, including Cisco, HP, Dell and Brocade. Moreover, and aside from their utility as USB connections, all the USB ports on these console servers can function as plain RS-232 serial ports when a USB-to-serial adapter is connected.

These USB ports are available as regular portmanager ports and are presented numerically in the web UI after all RJ-45 serial ports.

The LES1608A, for example, has eight RJ-45 serial ports on the rear of the console server and four USB ports on the front. In Serial & Network > Serial Port these are listed in the table below.

TABLE 5-3. RJ-45 AND USB PORTS ON THE LES1608A

PORT NUMBER	CONNECTOR
1	RJ-45
2	RJ-45
3	RJ-45
4	RJ-45
5	RJ-45
6	RJ-45
7	RJ-45
8	RJ-45
9	USB
10	USB
11	USB
12	USB

The common settings (baud rate etc.) are used when configuring the ports, but some operations (for example, sending serial

breaks) may not work depending on the implementation of the underlying USB serial chip.

5.1.10 LINK LAYER DISCOVERY PROTOCOL (LLDP)

The Link Layer Discovery Protocol (LLDP) is a protocol that allows system administrators to glean information about devices physically connected to managed switches. It is available for use on LES1700-R2, LES1516A, LES1532A, LES1548A and LES1600 devices.

The LLDP service is enabled through the System > Services page. When the service is enabled, the lldpd daemon is loaded and runs. The Service Access tab controls which network interfaces are monitored by the lldpd daemon.

When LLDP is granted access to an interface, it will use that interface even if the interface has been disabled via System > IP.

LLDP neighbors are visible through the Status > LLDP Neighbors page. This page shows neighbors heard, and also indicates the information that the console manager is sending.

NOTE: Although the LLDP service can be granted access to non-Ethernet interfaces (for example, G3, G4 and PSTN dial-up interfaces), it currently ignores non-Ethernet interfaces.

The lldpcli shell client interacts with and configures the running LLDP service.

Persistent custom configuration changes can be added to the system through configuration files placed in /etc/config/lldpd.d/. Custom configuration files—which must have filenames ending with .conf—will be read and executed by lldpcli when the LLDP service starts.

The /etc/ directory is read-only on Black Box hardware. Most default configuration files otherwise stored in /etc/ are, on Black Box hardware, in /etc/config/, which is writeable.

The default lldpd configuration file—lldpd.conf—is stored in /etc/config/ on Black Box hardware. It is not safe as a store of custom configuration details. There are circumstances in which this file is regenerated automatically, in which case all customizations will be lost.

The etc/config/lldpd.d/ directory, which is also writable and which is created on first boot, is safe to write to. Any Custom LLDP configurations must be stored as *.conf files in this directory.

When enabled, LLDP frames issued by an Black Box Console Manager will reveal sensitive information such as hostname, and firmware version.

LLDP frames are not passed through by 802.3ab compliant switches, and Black Box Console Managers have the LLDP service disabled by default.

Both lldpd and lldpcli have standard man pages but, because of space concerns, these pages are not shipped with Black Box hardware.

Both man pages are available on the lldpd project web-site, but: man lldpd is at https://vincentbernat.github.io/lldpd/usage.html#lldpd8; and man lldpcli is at https://vincentbernat.github.io/lldpd/usage.html#lldpcli8.

NOTE: Black Box uses lldpd 0.9.2.

5.2 ADD AND EDIT USERS

The Administrator uses this menu selection to set up, edit and delete users and to define the access permissions for each of these users.

Users can be authorized to access specified services, serial ports, power devices and specified network-attached hosts. These users can also be given full Administrator status (with full configuration and management and access privileges).

To simplify user set up, they can be configured as members of Groups. With firmware V3.5.2 and later, there are six Groups set up by default (earlier versions only had admin and user by default).

TABLE 5-4. USER GROUPS

GROUP	DESCRIPTION
admin	Provides users with unlimited configuration and management privileges.
pptpd	Group to allow access to the PPTP VPN server. Users in this group will have their password stored in clear text.
dialin	Group to allow access to the dialin server. Users in this group will have their password stored in clear text.
ftp	Group to allow ftp access and file access to storage devices.
pmshell	Group to set default shell to pmshell.
users	Provides users with basic management privileges.

Membership of the admin group provides the user with full Administrator privileges. The admin user (Administrator) can access the console server using any of the services that have been enabled in System: Services, e.g., if only HTTPS has been enabled, then the Administrator can only access the console server using HTTPS. Once logged in, they can reconfigure the console server settings (e.g., to enabled HTTP/Telnet for future access). They can also access any of the connected Hosts or serial port devices using any of the services that have been enabled for these connections. But again the Administrator can reconfigure the access services for any Host or serial port. So only trusted users should have Administrator access.

Membership of the user group provides the user with limited access to the console server and connected Hosts and serial devices. These Users can access only the Management section of the Management Console menu and they have no command line access to the console server. They also can only access those Hosts and serial devices that have been checked for them, using services that have been enabled.

If a user is set up with pptd, dialin, ftp or pmshell group membership, he will have restricted user shell access to the nominated managed devices but will not have any direct access to the console server itself. To add this, the user must also be a member of the "users" or "admin" groups.

The Administrator can also set up additional Groups with specific power device, serial port and host access permissions. Users in these additional groups don’t have any access to the Management Console menu nor do they have any command line access to the console server itself.

The Administrator can also set up users with specific power device, serial port and host access permissions, who are not a member of any Groups. Similarly, these users don’t have any access to the Management Console menu, nor do they have any command line access to the console server itself.

For convenience, the SDT Connector “Retrieve Hosts” function retrieves and auto-configures checked serial ports and checked hosts only, even for admin group users.

5.2.1 SETUP NEW GROUPS

To set up new Groups and new users, and to classify users as members of particular Groups:

- Select Serial & Network > Users & Groups to display the configured Groups and Users.
- Click Add Group to add a new Group.

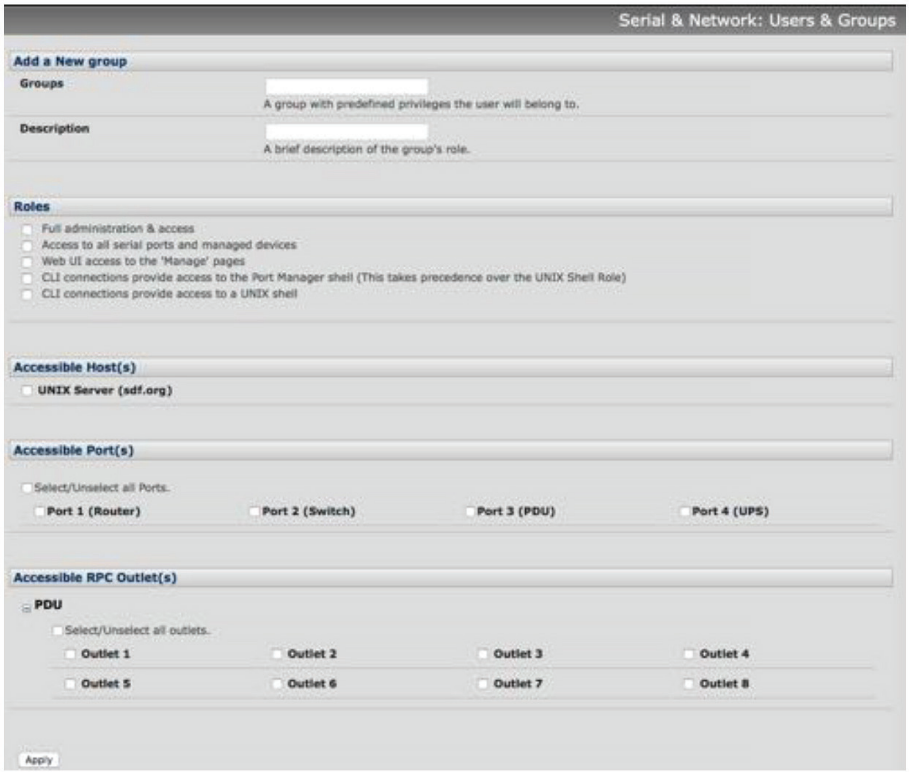


FIGURE 5-16. ADD NEW GROUP SCREEN

- Add a Group name and Description for each new Group, then nominate the Accessible Hosts, Accessible Ports and Accessible RPC Outlet(s) that you wish any users in this new Group to be able to access.
- Click Apply.
- The Administrator can Edit or Delete any added group.

5.2.2 SETUP NEW USERS

To set up new users, and to classify users as members of particular Groups:

- Select Serial & Network > Users & Groups to display the configured Groups and Users.
- Click Add User to add a new user.
- Add a Username for each new user. You may also include information related to the user (e.g. contact details) in the Description field.

NOTE: The User Name can contain from 1 to 127 alphanumeric characters as well as the following characters: - _ . (hyphen, underscore, and full-stop or period).

- Specify which Group (or Groups) you wish the user to be a member of.
- Add a confirmed Password for each new user.

NOTE: A user's Password can contain up to 254 characters. There are no restrictions on what characters are allowed in a password.

- SSH pass-key authentication can be used. This is more secure than password-based authentication. Paste the public keys of authorized public/private keypairs for this user in the Authorized SSH Keys field
- Check Disable Password Authentication if you wish to only allow public key authentication for this user when using SSH.
- Check Enable Dial-Back in the Dial-in Options menu to allow an out-going dial-back connection to be triggered by logging into this port.
- Enter the Dial-Back Phone Number to call-back when the user logs in.
- Check specific Accessible Hosts and Accessible Ports to nominate the serial ports and network connected hosts you wish the user to have access privileges to.
- If there are configured RPCs, you can check Accessible RPC Outlets to specify which outlets the user is able to control (that is, power on and off).
- Click Apply.

The new user will now be able to access the Network Devices, Ports and RPC Outlets you nominated as accessible plus, if the user is a Group member they can also access any other device/port/outlet that was set up as accessible to the Group

NOTE: There are no specific limits on user number; nor on the number of users per serial port or host. So multiple users (Users and Administrators) can control or monitor a port or host. Similarly, there are no specific limits on the group number and users can be a member of a number of Groups (and gain the cumulative access privileges of each Group). A user does not have to be a member of any Groups (but if the User is not even a member of the default user group then cannot use the Management Console to manage ports).

NOTE: While there are no specific limits, the time to re-configure does increase as the number and complexity increases. The aggregate number of users and groups should be kept under 250.

The Administrator can also edit the access settings for any existing users:

- Select Serial & Network > Users & Groups and click Edit to modify User access privileges.
- Alternatively click Delete to remove the user or Disable to temporarily block access.

NOTE: For more on enabling the SDT Connector so each user has secure tunneled remote RPD/VNC/Telnet/HHTP/HTTPS/SoL access to the network connected hosts, see Chapter 7.

5.3 AUTHENTICATION

See Chapter 10 for authentication configuration details.

5.4 NETWORK HOSTS

To monitor and remotely access a locally networked computer or device (referred to as a Host) identify the Host and specify the TCP or UDP ports/services used to control that Host.

- Select Serial & Network > Network Hosts.

All network-connected Hosts that have been enabled for access present as well as the related access TCP ports/services.

- Click Add Host to enable a new Host or select Edit to update an extant Host's settings.
- Enter the IP Address or the DNS Name and Host Name (up to 254 alphanumeric characters) for the new network connected Host.
- Enter a Description (this is an optional step).
- Add or edit the Permitted Services (or TCP/UDP port numbers) that are authorized to be used in controlling this host.

  Only these permitted services will be forwarded through by SDT to the Host. All other services (TCP/UDP ports) will be blocked.
- Set the Logging Level.

  This specifies the level of information to be logged and monitored for each Host access. See Chapter 8 for more information.
- If the Host is a PDU or UPS power device or a server with IPMI power control, specify RPC (for IPMI and PDU) or UPS and the Device Type.

  The Administrator can configure these devices and enable which users have permissions to remotely cycle power etc. (see Chapter 9). Otherwise, leave the Device Type set to None.

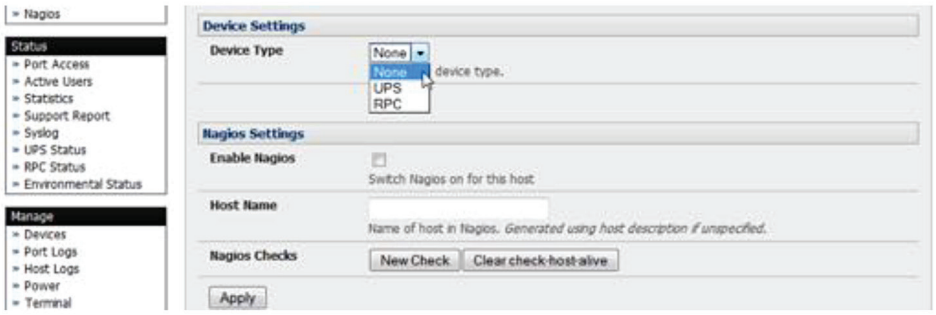


FIGURE 5-17. ENTER DEVICE TYPE

- If the console server has been configured with distributed Nagios monitoring enabled then you will also be presented with Nagios Settings options to enable nominated services on the Host to be monitored. See Chapter 11 for more information.
- Click Apply.

  This will create the new Host and also create a new Managed Device (with the same name).

5.5 TRUSTED NETWORKS

The Trusted Networks facility allows you an nominate specific IP addresses where users (Administrators and Users) must be located, to have access to console server serial ports:

- Select Serial & Network > Trusted Networks.
- Click Add Rule to add a new trusted network.

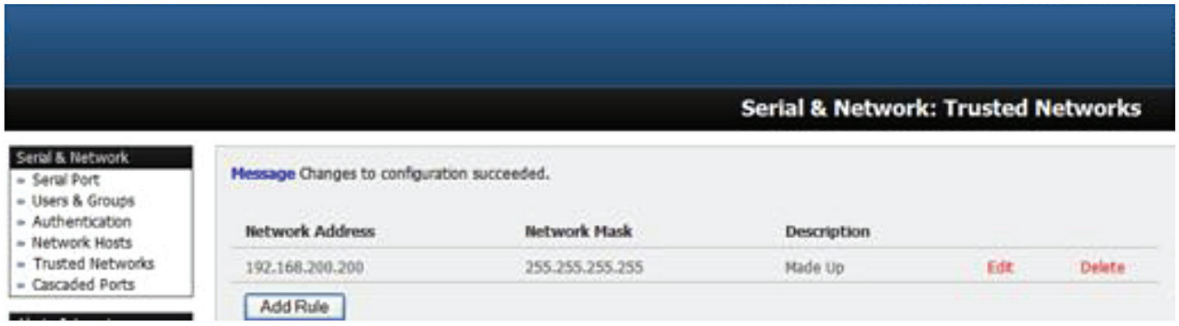


FIGURE 5-18. SERIAL & NETWORK: TRUSTED NETWORKS SCREEN, ADD RULE

NOTE: In the absence of Rules, there are no access limitations as to the IP address where Users or Administrators can be located.

- Select the Accessible Port(s) that the new rule is to be applied to.

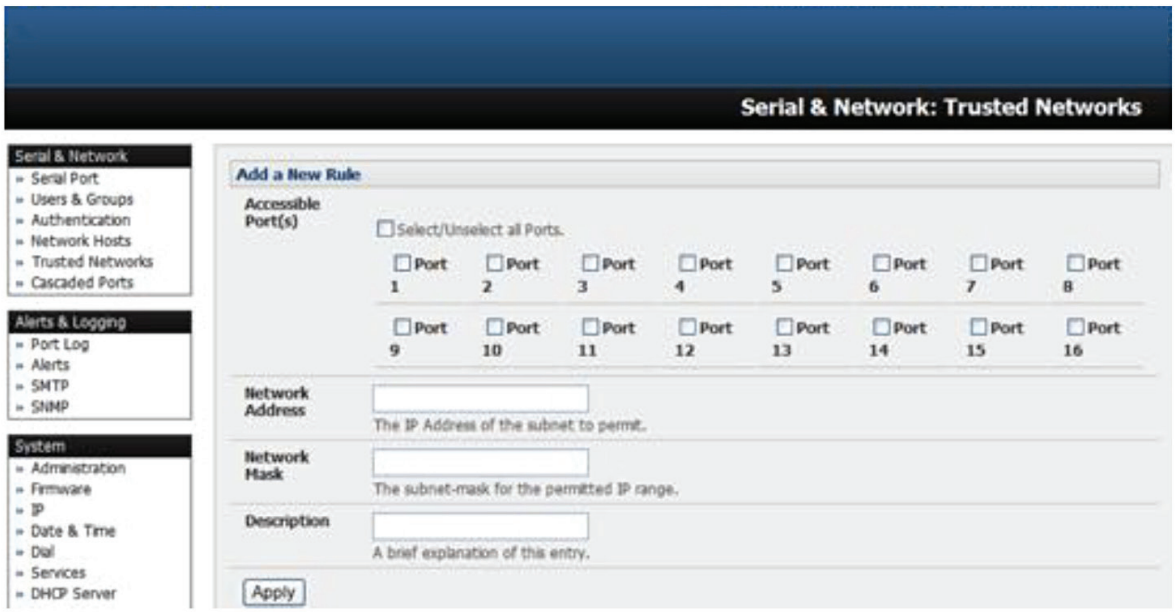


FIGURE 5-19. SERIAL & NETWORK: TRUSTED NETWORKS SCREEN

- Enter the Network Address of the subnet to be permitted access.
- Specify the range of addresses that are to be permitted by entering a Network Mask for that permitted IP range.
- For example, to permit all users located in the 204.15.5.0 Class C network to connect to the nominated port, would add the following Trusted Network rule:

  Network IP address: 204.15.5.0

  Subnet Mask: 255.255.255.0
- To permit only the user located at a specific IP address (in this case 204.15.5.13) to connect:

  Network IP address: 204.15.5.13

  Subnet Mask: 255.255.255.255

- To allow all users operating from within a specific range of IP addresses (in this case the 30 addresses from 204.15.5.129 to 204.15.5.158) to be permitted connection to the nominated port:

  Network IP address: 204.15.5.128

  Subnet Mask: 255.255.255.224

- Click Apply.

NOTE: The above Trusted Networks will limit access by Users and Administrators to the console serial ports. They do not restrict access by the Administrator to the console server itself or to attached hosts. To change the default settings for this access, you will to need to edit the IPtables rules as described in Chapter 16.

5.6 SERIAL PORT CASCADING

Cascaded Ports enables you to cluster distributed console servers so up to 1000 serial ports can be configured and accessed through one IP address and managed through the one Management Console. One console server, the Master, controls other console servers as Slave units and all the serial ports on the Slave units appear as if they are part of the Master.

Black Box's clustering connects each Slave to the Master with an SSH connection. This is done using public key authentication so the Master can access each Slave using the SSH key pair (rather than using passwords). This ensures secure authenticated communications between Master and Slaves enabling the Slave console server units to be distributed locally on a LAN or remotely around the world.

5.6.1 AUTOMATICALLY GENERATE AND UPLOAD SSH KEYS

To set up public key authentication, first generate an RSA or DSA key pair and upload them into the master and slave console servers. This can be done automatically from the Master.

- Select System > Administration on the master's Management Console.
- Check Generate SSH keys automatically.
- Click Apply.

Next, select whether to generate keys using RSA and/or DSA (if unsure, select only RSA).

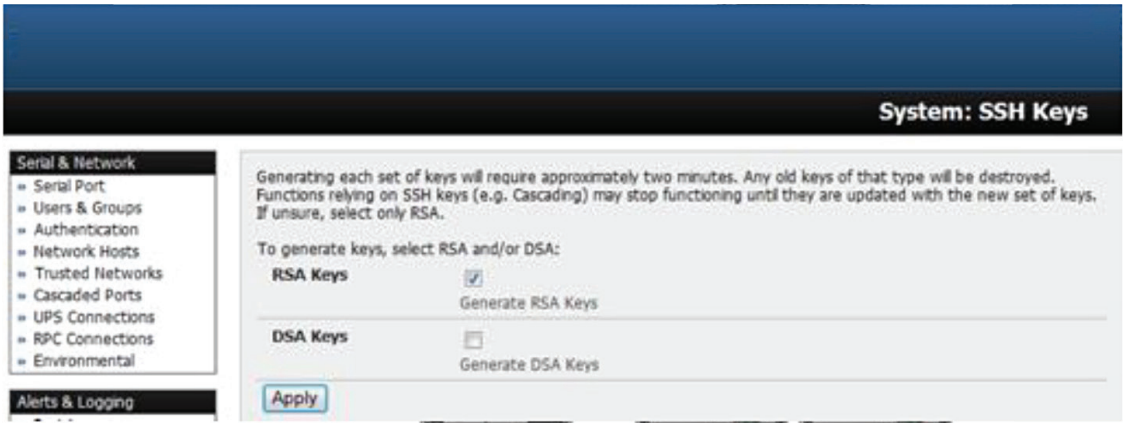


FIGURE 5-20. SSH KEYS SCREEN

Generating each set of keys will require approximately two minutes and the new keys will destroy any old keys of that type that may previously been uploaded. Also, while the new generation is underway on the master, functions relying on SSH keys (e.g. cascading) may stop functioning until they are updated with the new set of keys. To generate keys:

- Check RSA Keys, DSA Keys, or both.
- Click Apply.
- Once the new keys have been generated, Click here to return and the keys will automatically be uploaded to the master and connected slaves.

5.6.2 MANUALLY GENERATE AND UPLOAD SSH KEYS

To manually upload the key public and private key pair to the Master console server:

- Select System > Administration on the master's Management Console.
- Browse to the location you have stored RSA (or DSA) Public Key and upload it to SSH RSA (DSA) Public Key.
- Browse to the stored RSA (or DSA) Private Key and upload it to SSH RSA (DSA) Private Key.
- Click Apply.

Next, you must register the Public Key as an Authorized Key on the slave. In the simple case with only one master with multiple slaves, you need only upload the one RSA or DSA public key for each slave.

NOTE: The use of key pairs can be confusing as in many cases one file (Public Key) fulfills two roles—Public Key and Authorized Key. For a more detailed explanation see Authorized in Section 16.6. Also refer to this chapter if you need to use more than one set of Authorized Keys in the slave.

- Select System > Administration on the slave's Management Console.

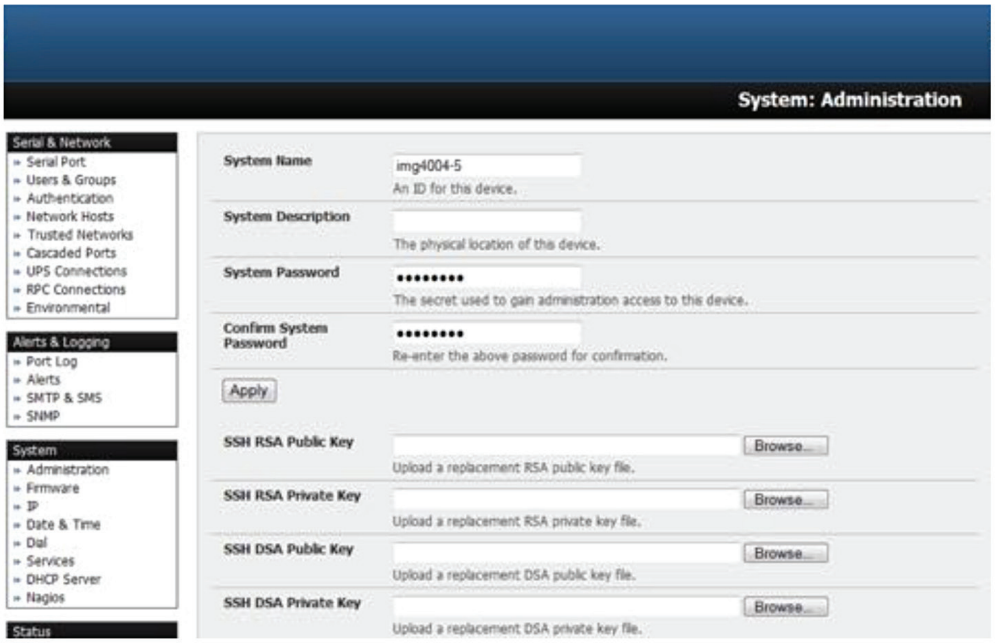


FIGURE 5-21. SYSTEM ADMINISTRATION SCREEN

Browse again to the stored RSA (or DSA) Public Key and upload it to Slave's SSH Authorized Key.

- Click Apply.

The next step is to Fingerprint each new slave-master connection. This once-off step will validate that you are establishing an SSH session to who you think you are. On the first connection the Slave will receive a fingerprint from the Master which will be used on all future connections.

- Log in to the master console server as root.
- Establish an SSH connection to the remote slave host:

```
# ssh remote-host-name
```

Once the SSH connection has been established, you will be asked to accept the key. Answer yes and the fingerprint will be added to the list of known hosts. For more details on Fingerprinting see Section 16.6.

NOTE: If you are asked to supply a password, then there is a problem with uploading keys. The keys should remove any need to supply a password.

5.6.3 CONFIGURE THE SLAVES AND THEIR SERIAL PORTS

You can now begin setting up the slaves and configuring slave serial ports from the master console server.

- Select Serial & Network > Cascaded Ports on the master's Management Console.

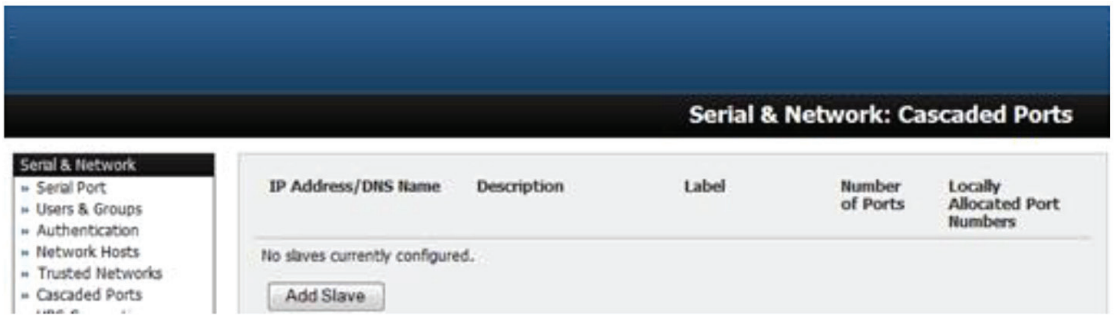


FIGURE 5-22. SERIAL & NETWORK: CASCADED PORTS SCREEN

- Click Add Slave to add clustering support.

NOTE: You cannot add any slaves until you have automatically or manually generated SSH keys.

To define and configure a slave:

- Enter the remote IP Address (or DNS Name) for the Slave console server.
- Enter a brief Description and a short Label for the slave.

  Use a convention here that enables effective management of large networks of clustered console servers and the connected devices.
- Enter the full number of serial ports on the slave unit in Number of Ports.
- Click Apply.

  This will establish the SSH tunnel between the master and the new slave.

The Serial & Network > Cascaded Ports menu displays all the slaves and the port numbers that have been allocated on the master. If the master console server has 16 ports of its own, then ports 1–16 are pre-allocated to the master, so the first slave added will be assigned port number 17 onwards.

Once you have added all the slave console servers, the slave serial ports and the connected devices are configurable and accessible from the master's Management Console menu and accessible through the Master's IP address.

- Select the appropriate Serial & Network > Serial Port and Edit to configure the serial ports on the slave.
- Select the appropriate Serial & Network > Users & Groups to add new users with access privileges to the slave serial ports (or to extend existing users access privileges).
- Select the appropriate Serial & Network > Trusted Networks to specify network addresses that can access nominated slave serial ports.
- Select the appropriate Alerts & Logging > Alerts to configure slave port Connection, State Change or Pattern Match alerts.
- Click Apply.

  The configuration changes made on the master are propagated out to all the Slaves.

5.6.4 MANAGING THE SLAVES

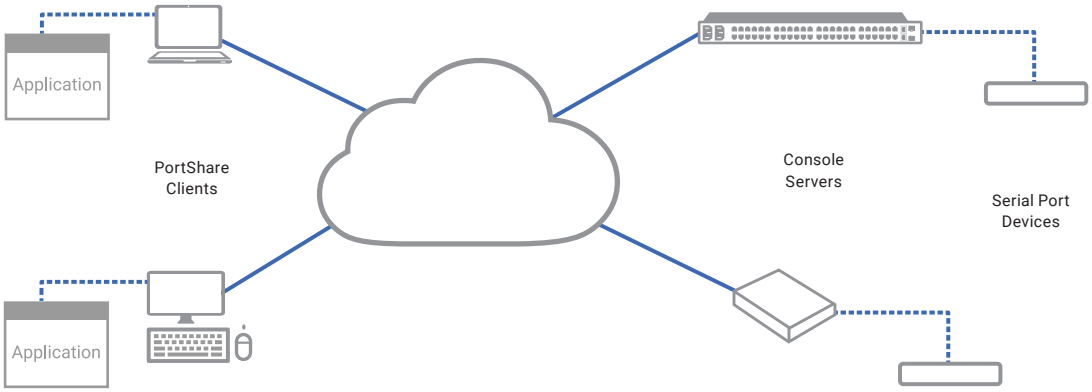


FIGURE 5-23. SLAVE CONFIGURATION

The master is in control of the slave serial ports. So, for example, if you change a User access privileges or edit any serial port setting on the master, the updated configuration files will be sent out to each slave in parallel. Each slave will then automatically make changes to their local configurations (and only make those changes that relate to its particular serial ports).

You can still use the local slave Management Console to change the settings on any slave serial port (such as alter the baud rates). These changes will be overwritten the next time the master sends out a configuration file update.

While the master is in control of all slave serial port related functions, it is not master over the slave network host connections or over the slave console server system itself.

So, slave functions such as IP, SMTP & SNMP Settings, Date &Time, DHCP server must be managed by accessing each slave directly and these functions are not overwritten when configuration changes are propagated from the master. Similarly, the slaves Network Host and IPMI settings have to be configured at each slave.

The master's Management Console provides a consolidated view of the settings for its own and the entire slave's serial ports, but the master does not provide a fully consolidated view. For example if you want to find out who's logged in to cascaded serial ports from the master, you'll see that Status > Active Users only displays those users active on the master's ports, so you may need to write custom scripts to provide this view. This is covered in Chapter 12.

5.7 SERIAL PORT REDIRECTION (PORTSHARE)

PortShare 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 PortShare on one or more computers for accessing any serial device connected to a console server port.

PortShare for Windows

The portshare_setup.exe program is included on the CD supplied with your console server. A copy can be freely downloaded from the ftp site. Refer to the PortShare User Manual and Quick Start for details on installation and operation.

PortShare for Linux

The PortShare driver for Linux maps the console server serial port to a host tty port. Black Box has released the portshare-serial-client as an open source utility for Linux, AIX, HPUX, SCO, Solaris and UnixWare. This utility can be freely downloaded from the ftp site.

The 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 network and receives data from the console server through network and transmits it to the pseudo-tty port.

The .tar file can be freely downloaded from the ftp site. Refer to the PortShare User Manual and Quick Start for details on installation and operation.

5.8 MANAGED DEVICES

Managed Devices presents a consolidated view of all the connections to a device that can be accessed and monitored through the console server. To view the connections to the devices:

- Select Serial & Network > Managed Devices.

This screen displays all the Managed Device with their Description, Notes and lists of all the configured Connections.

- Serial Port #: if serially connected.
- USB: if USB connected.
- IP Address: if network connected.
- Power PDU/outlet: if applicable.
- UPS connections: if applicable.

Devices such as servers will commonly have more than one power connection and more than one network connection (for example, for BMC/service processor).

All users can view (but not edit) these Managed Device connections by selecting Manage > Devices. The Administrator can edit, add to and delete Managed Devices and connections.

To edit an existing device and add a new connection:

- Select Serial & Network > Managed Devices.
- Click Edit.
- Click Add Connection.
- Select the connection type for the new connection (Serial, Network Host, UPS or RPC).
- Select the specific connection from the presented list of configured unallocated hosts/ports/outlets.

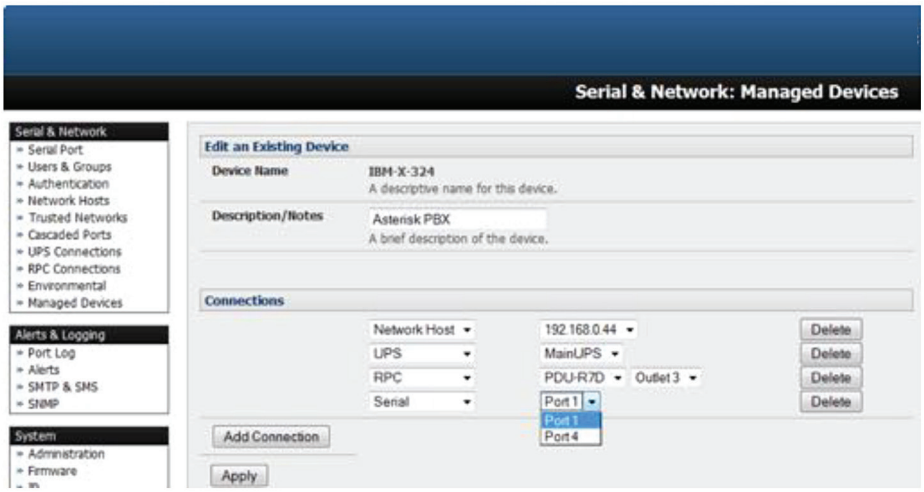


FIGURE 5-24. EDIT AN EXISTING DEVICE SCREEN

To add a new network connected Managed Device:

- The Administrator adds a new network connected Managed Device using Add Host on the Serial & Network > Network Host menu. This automatically creates a corresponding new Managed Device (as covered in Section 5.4).
- When adding a new network connected RPC or UPS power device, you set up a Network Host, designate it as RPC or UPS, then go to RPC Connections (or UPS Connections) to configure the relevant connection.
- A corresponding new Managed Device (with the same Name and Description as the RPC/UPS Host) is not created until this connection step is completed (see Chapter 10).

NOTE: The outlet names on a newly created PDU will, by default, be “Outlet 1” and “Outlet 2.” When you connect a particular Managed Device (that draws power from the outlet) the outlet will take up the name of the powered Managed Device.

To add a new serially connected Managed Device:

- Configure the serial port using the Serial & Network > Serial Port menu (see Section 5.1).

Select Serial & Network > Managed Devices.

- Click Add Device.
- Enter a Device Name and Description for the Managed Device.
- Click Add Connection and select Serial and the Port that connects to the Managed Device.
- Click Add Connection to add a UPS/RPC power connection or network connection or another serial connection.
- Click Apply.

NOTE: To set up a new serially connected RPC UPS or EMD device, you configure the serial port, designate it as a Device then enter a Name and Description for that device in the Serial & Network: RPC Connections (or UPS Connections or Environmental). When applied,
this will automatically create a corresponding new Managed Device with the same Name and Description as the RPC/UPS Host (see Chapter 9).

NOTE: The outlet names on the PDU will, by default, be “Outlet 1” and “Outlet 2.” When you connect a particular Managed Device (that draws power from the outlet) the outlet will take up the name of the powered Managed Device.

5.9 IPSEC VLAN

The LES1600, LES1516A, LES1532A, LES1548A, and LES1700-R2 family of advanced console servers include a Linux implementation of the IPsec (IP Security) protocols, which can be used to configure a Virtual Private Network (VPN). The VPN allows multiple sites or remote administrators to access the Black Box advanced console server (and Managed Devices) securely over the Internet.

The administrator can establish encrypted authenticated VPN connections between advanced console servers distributed at remote sites and a VPN gateway (such as Cisco router running IOS IPsec) on their central office network.

Users and administrators at the central office can then securely access the remote console servers and connected serial console devices and machines on the Management LAN subnet at the remote location as though they were local.

With serial bridging, serial data from controller at the central office machine can be securely connected to the serially controlled devices at the remote sites (see Section 5.1).

The road warrior administrator can use a VPN IPsec software client such as TheGreenBow (https://thegreenbow.com/) or Shrew Soft (https://shrew.net/) to remotely access the advanced console server and every machine on the Management LAN subnet at the remote location.

Configuration of IPsec is quite complex so Black Box provides a simple GUI interface for basic set up as described below.

ENABLE THE VPN GATEWAY

- Select Serial & Networks > IPsec VPN.
- Click Add.
- Complete the Add IPsec Tunnel screen.
- Enter a descriptive name to identify the added IPsec Tunnel. For example West-St-Outlet.
- Select the Authentication Method: either RSA digital signatures or a Shared secret (PSK).
- If you select RSA, you will asked to click here to generate keys. This will generate an RSA public key for the console server (the Left Public Key). You will need to find out the key to be used on the remote gateway, then cut and paste it into the Right Public Key.
- If you select Shared secret, you will need to enter a Pre-shared secret (PSK). The PSK must match the PSK configured at the other end of the tunnel.
- In Authentication Protocol, select the authentication protocol to be used. Either authenticate as part of ESP (Encapsulating Security Payload) encryption or separately using the AH (Authentication Header) protocol.
- Enter a Left ID and Right ID. This is the identifier that the Local host/gateway and remote host/gateway use for IPsec negotiation and authentication.
- Each ID must include an @ and can include a fully qualified domain name preceded by @ (for example, left@example.com).

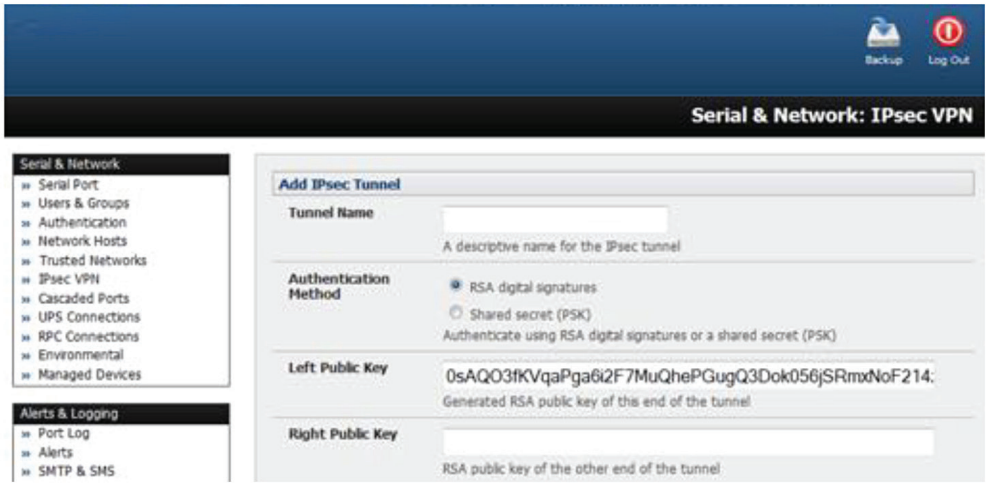


FIGURE 5-25. ADD IPSEC TUNNEL SCREEN

Enter the public IP or DNS address of this Black Box VPN gateway as the Left Address. You can leave this blank to use the interface of the default route.

- In Right Address, if the remote end has a static or dyndns address, enter the public IP or DNS address of the remote end of the tunnel. Otherwise, leave this blank.
- If the Black Box VPN gateway serves as a VPN gateway to a local subnet (e.g., the console server has a Management LAN configured) enter the private subnet details in Left Subnet.

  Use the CIDR notation, where the IP address number is followed by a slash and the number of 'one' bits in the binary notation of the netmask.

  For example 192.168.0.0/24 indicates an IP address where the first 24 bits are used as the network address. This is the same as 255.255.255.0.

  If the VPN access is only to the console server itself and to its attached serial console devices then leave Left Subnet blank.
- If there is a VPN gateway at the remote end, enter the private subnet details in Right Subnet.

  Again use CIDR notation and leave blank if there is only a remote host.
- Select Initiate Tunnel if the tunnel connection is to be initiated from the Left console server end.

  This can only be initiated from the VPN gateway (Left) if the remote end was configured with a static (or dyndns) IP address.
- Click Apply to save changes.

NOTE: It is essential the configuration details set up on the advanced console server (referred to as the Left or Local host) exactly matches the set up entered when configuring the Remote (Right) host/gateway or software client.

5.10 OPENVPN

The LES1600, LES1516A, LES1532A, LES1548A, and LES1700-R2 family of advanced console servers with Firmware v3.2 and later, include OpenVPN. OpenVPN uses the OpenSSL library for encryption, authentication, and certification, which means it uses SSL/TSL (Secure Socket Layer/Transport Layer Security) for key exchange and can encrypt both data and control channels. Using OpenVPN allows
for the building of cross-platform, point-to-point VPNs using either X.509 PKI (Public Key Infrastructure) or custom configuration files.

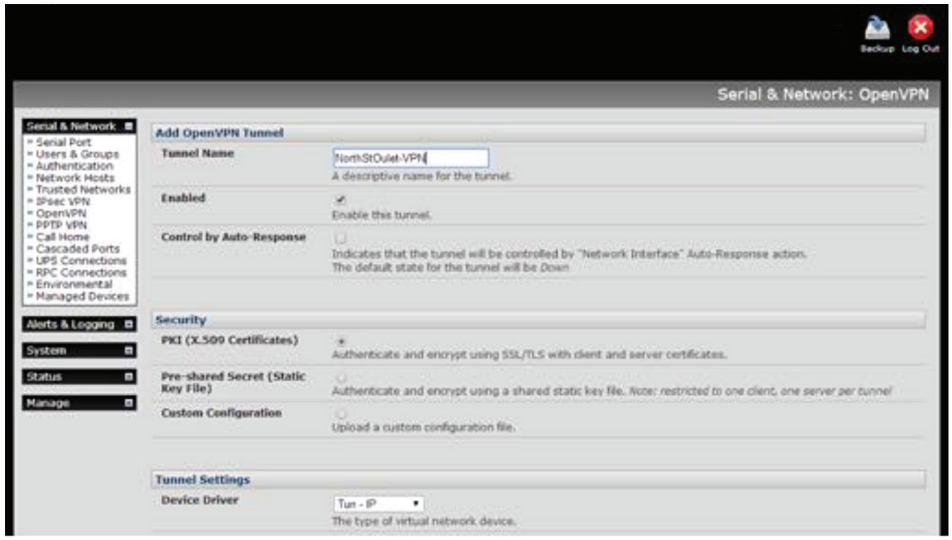


FIGURE 5-26. OPENVPN SCREEN

OpenVPN allows secure tunneling of data through a single TCP/UDP port over an unsecured network, thus providing secure access to multiple sites and secure remote administration to a console server over the Internet.

OpenVPN also allows the use of Dynamic IP addresses by both the server and client thus providing client mobility. For example, an OpenVPN tunnel may be established between a roaming windows client and an Black Box advanced console server within a data center.

Configuration of OpenVPN can be complex so Black Box provides a simple GUI interface for basic set up as described next.

5.10.1 ENABLE THE OPENVPN

Select Serial & Networks > OpenVPN.

- Click Add.
- Fill-out the required fields on the Add OpenVPN Tunnel screen.
- Enter a descriptive name to identify the added IPsec Tunnel. For example West-St-Outlet.
- Select the authentication method to be used.

  To authenticate using certificates, select PKI (X.509 Certificates).

  To authenticate using a custom configuration, select Custom Configuration to upload custom configuration files.

NOTE: Custom configurations must be stored in /etc/config.

If you select PKI (public key infrastructure), you will need to establish:

- a separate certificate (also known as a public key).

  This Certificate File will be a *.crt file type.

- a Private Key for the server and each client.

  This Private Key File will be a *.key file type.
- A master Certificate Authority (CA) certificate and key which is used to sign each of the server and client certificates.

  This Root CA Certificate will be a *.crt file type.

For a server, you may also need dh1024.pem (Diffie Hellman parameters).

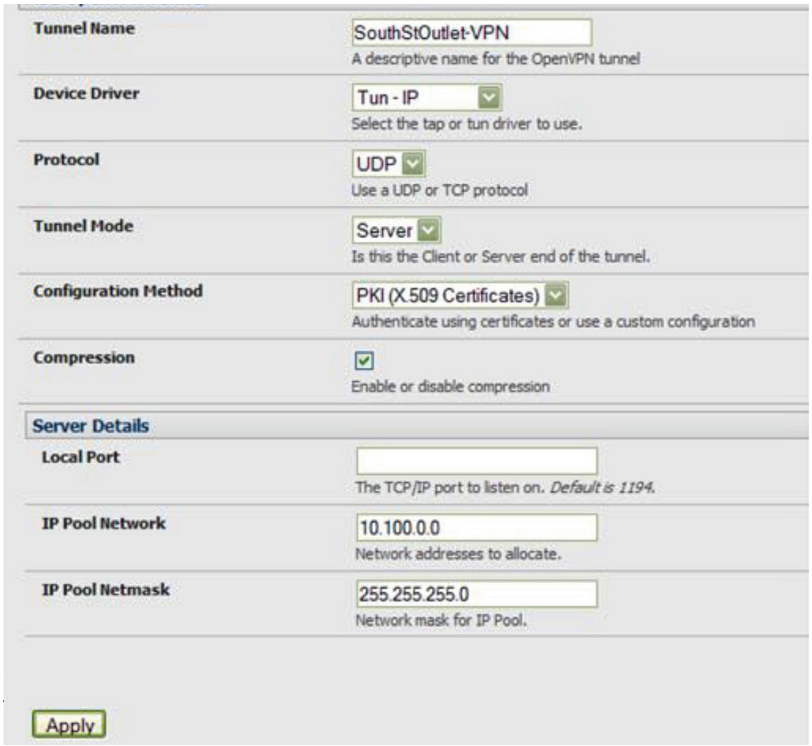


FIGURE 5-27. SERVER DETAILS SCREEN

See http://openvpn.net/easyrsa.html for a guide to basic RSA key management. For alternative authentication methods see http://openvpn.net/index.php/documentation/howto.html#auth. For more information also see http://openvpn.net/howto.html.

- Select the Device Driver to be used, either Tun-IP or Tap-Ethernet.

  The TUN (network tunnel) and TAP (network tap) drivers are virtual network drivers that support IP tunneling and Ethernet tunneling, respectively. TUN and TAP are part of the Linux kernel.
- Select either UDP or TCP as the Protocol.

  UDP is the default and preferred protocol for OpenVPN.
- In Tunnel Mode, nominate whether this is the Client or Server end of the tunnel.

  When running as a server, the advanced console server supports multiple clients connecting to the VPN server over the same port.
- Check or uncheck the Compression button to enable or disable compression.

5.10.2 CONFIGURE AS SERVER OR CLIENT

- Complete the Client Details or Server Details depending on the Tunnel Mode selected.

  If Client is selected, the Primary Server Address will be the address of the OpenVPN Server.

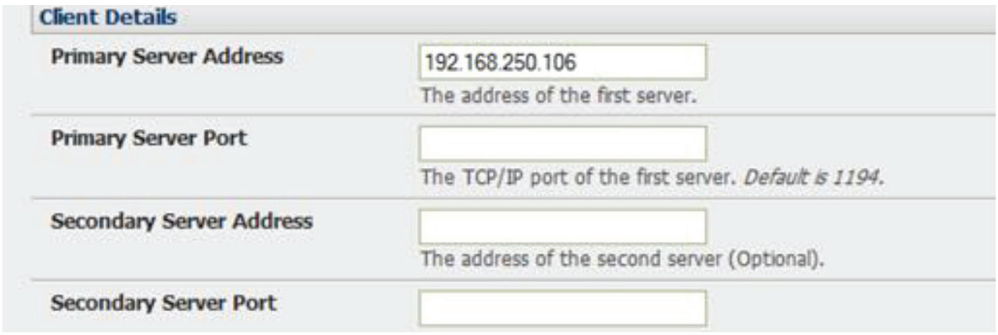


FIGURE 5-28. CLIENT DETAILS SCREEN

If Server is selected, enter the IP Pool Network address and the IP Pool Network mask for the IP Pool. The IP Pool Network provides addresses for connecting clients.

- Click Apply.
- To enter authentication certificates and files, Edit the OpenVPN tunnel.

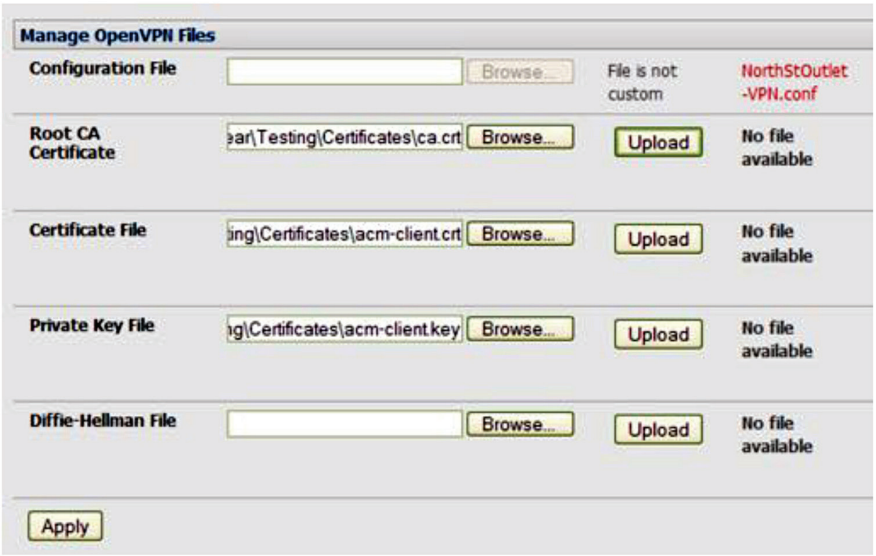


FIGURE 5-29. MANAGE OPENVPN FILES SCREEN

- Select the Manage OpenVPN Files tab. Upload or browse to relevant authentication certificates and files.
- Click Apply.

  Saved files will be displayed in red to the right-hand side of the Upload button.

FIGURE 5-30. SAVED FILES DISPLAYED ON SCREEN

- To enable OpenVPN, Edit the OpenVPN tunnel.
- Check the Enabled checkbox.
- Click Apply.
- Select Status > Statistics to verify that the tunnel is operational.

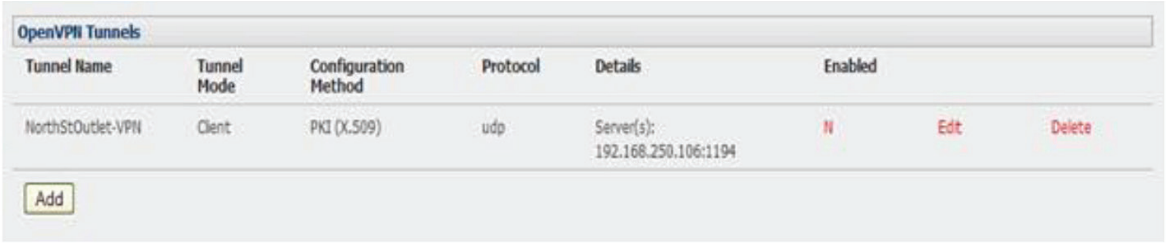


FIGURE 5-31. STATISTICS

NOTE: The console server system time must be correct, otherwise, authentication issues can arise.

5.10.3 WINDOWS OPENVPN CLIENT AND SERVER SETUP

Windows does not come standard with any OpenVPN server or client. This section outlines the installation and configuration of a Windows OpenVPN client or a Windows OpenVPN server and setting up a VPN connection to a console server.

Console servers with firmware V3.5.2 and later will generate Windows client config automatically from the GUI for Pre-shared Secret (Static Key File) configurations.

Alternately, OpenVPN GUI for Windows software (which includes the standard OpenVPN package plus a Windows GUI) can be downloaded from https://openvpn.net/.

Once installed on the Windows machine, an OpenVPN icon will present in the Notification Area located in the right side of the taskbar.

- Right click on this icon to start (and stop) VPN connections, and to edit configurations and view logs.

When the OpenVPN software is started, the C:\Program Files\OpenVPN\config folder will be scanned for .opvn files. This folder is rechecked for new configuration files whenever the OpenVPN GUI icon is right-clicked.

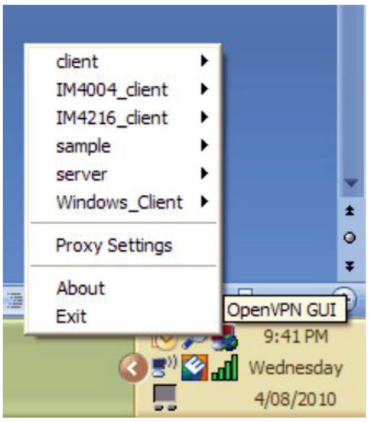


FIGURE 5-32. OPENVPN GUI ICON

So once the OpenVPN client is installed, a configuration file will need to be created.

- Using a text editor, create an xxxx.ovpn file and save in C:\Program Files\OpenVPN\config\. For example, C:\Program Files\OpenVPN\config\client.ovpn.
- An example OpenVPN Windows client configuration file:

  # description: LES1416A_client

  client

  proto udp

  verb 3

  dev tun

  remote 192.168.250.152

  port 1194

  ca c:\\openvpnkeys\\ca.crt

  cert c:\\openvpnkeys\\client.crt

  key c:\\openvpnkeys\\client.key

  nobind

  persist-key

  persist-tun

  comp-lzo
- An example OpenVPN Windows server configuration file:

  server 10.100.10.0 255.255.255.0

  port 1194

  keepalive 10 120

  proto udp

  mssfix 1400

  persist-key

  persist-tun

dev tun

ca c:\\openvpnkeys\\ca.crt

cert c:\\openvpnkeys\\server.crt

key c:\\openvpnkeys\\server.key

dh c:\\openvpnkeys\\dh.pem

comp-lzo

verb 1

syslog LES1416A_OpenVPN_Server

The Windows client/server configuration file options are listed in the next table:

TABLE 5-5. WINDOWS CLIENT/SERVER CONFIGURATION FILE OPTIONS

OPTION	DESCRIPTION
# comments and notes	Lines beginning with # are ignored by OpenVPN.
client or server	Specify whether this will be a client or server configuration file. In the server configuration file, define the IP address pool and netmask. For example: server 10.100.10.0 255.255.255.0
proto [udp | tcp]	Set the protocol. Client and server must be the same.
mssfix size	Set a packet's maximum size. Only useful for UDP if problems occur.
verb level	Set log-file verbosity. Values range from 0–15. 0 = silent except for fatal errors. 3 = medium output logging. Good for general use. 5 = helps with debugging connection problems. 9 = extremely verbose. Excellent for troubleshooting.
dev [tun | tap]	Set dev tun to create a routed IP tunnel. Set dev tup to create an Ethernet tunnel. Client and server must be the same.
remote host	Set the hostname or IP address of the OpenVPN server. Mandatory but a client-only setting.
Port	The UDP or TCP port of the OpenVPN server.
Keepalive ping-value down-value	Uses ping to keep the OpenVPN session alive. For example: Keepalive 10 120 pings the server every ten seconds and assumes the remote peer is down if no ping is received after 120 seconds (two minutes).
ca file-name	Enter the CA certificate file name and location The same CA certificate can be used by the server and all clients. Ensure each \ in the directory path is escaped. For example: c:\openvpnkeys\ca.crt must be entered as: c:\\openvpnkeys\\ca.crt

TABLE 5-5 (CONTINUED). WINDOWS CLIENT/SERVER CONFIGURATION FILE OPTIONS

OPTION	DESCRIPTION
cert file-name	Enter the client's or server's certificate file name and location Each client should have its own certificate and key files. As above, each \ in the directory path must be escaped.
key file-name	Enter the client's or server's key file name and location Each client should have its own certificate and key files. As above, each \ in the directory path must be escaped.
dh file-name	Enter the path to the key with the Diffie-Hellman parameters. A server-only setting.
Nobind	Used when clients do not need to bind to a local address or specific local port number. This is the case in most client configurations.
persist-key	Prevents the reloading of keys across restarts.
persist-tun	Prevents the closing and reopening of TUN/TAP devices across restarts.
cipher [BF-CBC Blowfish | AES-128-CBC AES | DES-EDE3-CBC Triple DES]	Sets the cryptographic cipher. BF-CBC Blowfish is the default if no cipher is explicitly set. The client and server must use the same settings.
comp-lzo	Enables compression on the OpenVPN link. If enabled, it must be set on the client and the server.
syslog	Located in syslog on Linux or Unix. Located in \Program Files\OpenVPN\log\ if running as a service on Windows.

To initiate the OpenVPN tunnel following the creation of the client/server configuration files:

- Right click on the OpenVPN icon in the Notification Area.
- Select the newly created client or server configuration.
- Click Connect in the presented sub-menu.
- The log file will display as the connection is established.
- Once established, the OpenVPN icon will display a message notifying of the successful connection and assigned IP.

This information, as well as the time the connection was established, is available anytime by scrolling over the OpenVPN icon.

NOTE: An alternate, open-source OpenVPN Windows client can be downloaded from https://openvpn.net/index.php/open-source/downloads.html. See https://openvpn.net/index.php/access-server/docs for help.

FIGURE 5-33. OPENVPN WINDOWS CLIENT

5.11 PPTP VPN

The LES1600, LES1516A, LES1532A, LES1548A, and LES1700-R2 family of advanced console servers with firmware v3.5.2 and later, include a PPTP (Point-to-Point Tunneling Protocol) server.

PPTP is typically used for communications over a physical or virtual serial link. The PPP endpoints define a virtual IP address to themselves. Routes to networks can then be defined with these IP addresses as the gateway, which results in traffic being sent across the tunnel. PPTP establishes a tunnel between the physical PPP endpoints and securely transports data across the tunnel.

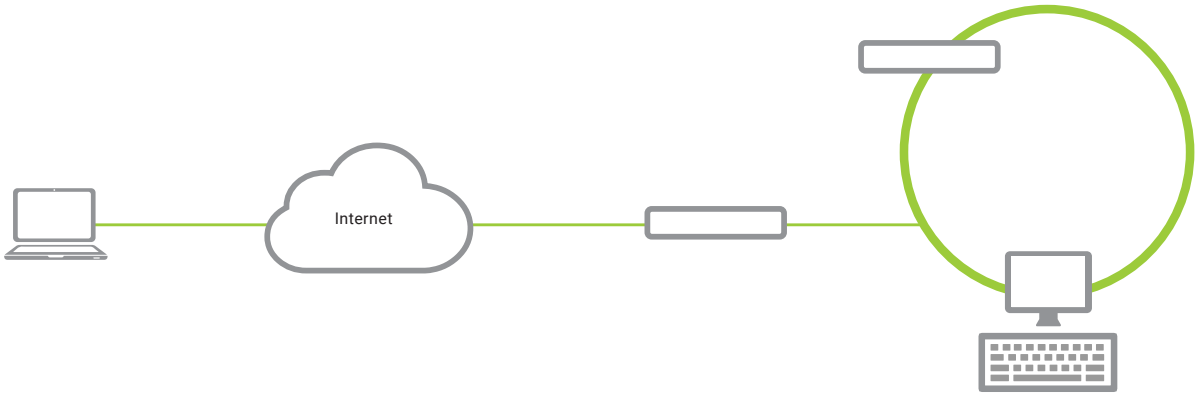


FIGURE 5-34. PPTP VPN

The strength of PPTP is its ease of configuration and integration into existing Microsoft infrastructure. It is generally used for connecting single remote Windows clients.

If you take your portable computer on a business trip, you can dial a local number to connect to your Internet access service provider (ISP) and then create a second connection (tunnel) into your office network across the Internet and have the same access to your corporate network as if you were connected directly from your office. Similarly, telecommuters can also set up a VPN tunnel over their cable modem or DSL links to their local ISP.

To set up a PPTP connection from a remote Windows client to your Black Box appliance and local network:

- Enable and configure the PPTP VPN server on your Black Box appliance.
- Set up VPN user accounts on the Black Box appliance and enable the appropriate authentication.
- Configure the VPN clients at the remote sites. The client does not require special software as the PPTP Server supports the standard PPTP client software included with Windows NT and later.
- Connect to the remote VPN.

5.11.1 ENABLE THE PPTP VPN SERVER

- Select PPTP VPN on the Serial & Networks menu.
- Click the Enable check box to enable the PPTP Server.
- Select the Minimum Authentication Required.

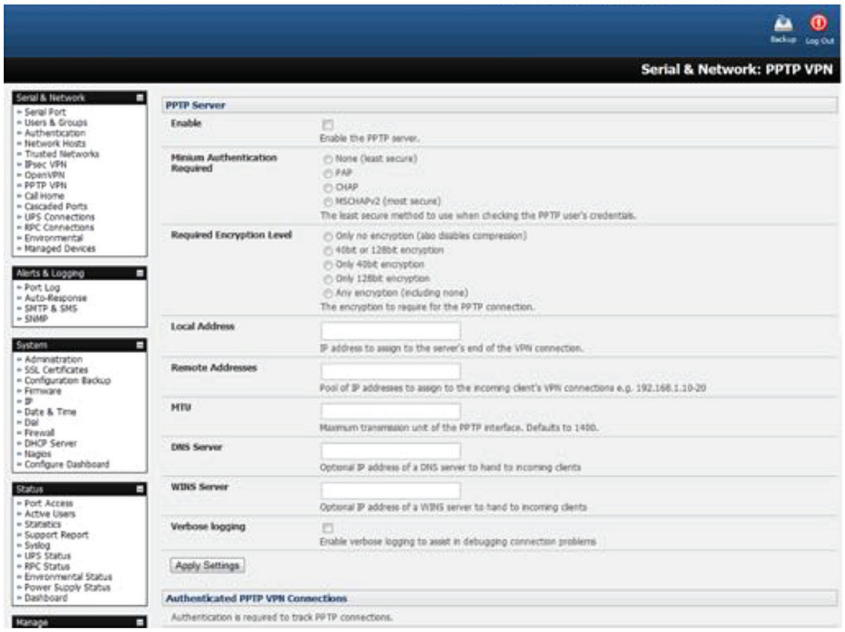


FIGURE 5-35. SERIAL & NETWORK: PPTP VPN SCREEN

Access is denied to remote users attempting to connect using an authentication scheme weaker than the selected scheme. From strongest to weakest, the schemes are:

- Encrypted Authentication (MS-CHAP v2). The strongest and recommended authentication option.
- Weakly Encrypted Authentication (CHAP). This is the weakest type of encrypted password authentication to use. It is not recommended that clients connect using this as it provides very little password protection. Also note that clients connecting using CHAP are unable to encrypt traffic.
- Unencrypted Authentication (PAP). This is plain text password authentication. When using this type of authentication, the client password is transmitted unencrypted.
- None. No encryption at all.

- Select the Required Encryption Level.

  Access is denied to remote users attempting to connect not using this encryption level. 40-bit or 128-bit encryption is recommended.

- In Local Address enter the IP address to assign to the server's end of the VPN connection.
- In Remote Addresses enter the pool of IP addresses to assign to the incoming client's VPN connections (for example, 192.168.1.10-20).

  These must be free IP addresses, from the network (typically the LAN) that remote users are assigned while connected to the Black Box appliance.

- Enter the desired value of the Maximum Transmission Unit (MTU) for the PPTP interfaces into the MTU field (defaults to 1400).
- In the DNS Server field, enter the IP address of the DNS server that assigns IP addresses to connecting PPTP clients.

- In the WINS Server field, enter the IP address of the WINS server that assigns IP addresses to connecting PPTP client.
- Enable Verbose Logging to assist in debugging connection problems.
- Click Apply.

5.11.2 ADD A PPTP USER

- Navigate to Serial & Networks > Users & Groups.
- Complete the fields as covered in Section 5.2.
- Ensure the pptpd Group has been checked, to allow access to the PPTP VPN server.

NOTE: Users in this group will have their password stored in clear text.

- Note the username and password for when you connect to the VPN connection.
- Click Apply.

5.11.3 SETUP A REMOTE PPTP CLIENT

Ensure the remote VPN client PC has Internet connectivity. To create a VPN connection across the Internet, you must set up two networking connections. One connection is for the ISP, and the other connection is for the VPN tunnel to the Black Box appliance.

NOTE: This procedure sets up a PPTP client under Windows 7 Professional. The steps may vary slightly depending on your network access or if you are using a different version of Windows.

- Login to your Windows system with administrator privileges.

FIGURE 5-36. CHOOSE A CONNECTION OPTION SCREEN

- From the Network & Sharing Center in the Control Panel select Network Connections and create a new connection.
- Select Use My Internet Connection (VPN) and enter the IP Address of the Black Box appliance.

NOTE: To connect remote VPN clients to the local network, you need to know the user name and password for the PPTP account you added, as well as the Internet IP address of the Black Box appliance. If your ISP has not allocated you a static IP address, consider using a dynamic DNS service. Otherwise, you must modify the PPTP client configuration each time your Internet IP address changes.

5.12 CALL HOME

Console servers with firmware v3.2 and later include Call Home. Call Home sets up an SSH tunnel from the console server to a central Virtual Central Management System (VCMS) server (referred to herein as VCMS). The console server then registers as a candidate on the VCMS. Once accepted it becomes a Managed Console Server.

The VCMS will then monitor the Managed Console Server, and administrators can access the remote Managed Console Server, through the VCMS. This access is available even when the remote console server is behind a third party firewall or has a private IP addresses (which is often the case when the console server is connected via a cellular modem connection).

VCMS maintains public key authenticated SSH connections to each Managed Console Server. These connections are used for monitoring, commanding and accessing the Managed Console Servers and the Managed Devices connected to the Managed Console Server.

To manage Local Console Servers, or console servers that are reachable from the VCMS, the SSH connections are initiated by VCMS. To manage Remote Console Servers, or console servers that are firewalled, not routable, or otherwise unreachable from the VCMS, the SSH connections are initiated by the Managed Console Server via an initial Call Home connection.

This ensures secure, authenticated communications and enables Managed Console Servers units to be distributed locally on a LAN, or remotely around the world.

5.12.1 SET UP CALL HOME CANDIDATE

To set up the console server as a Call Home management candidate on the VCMS:

- Select Call Home on the Serial & Network menu.

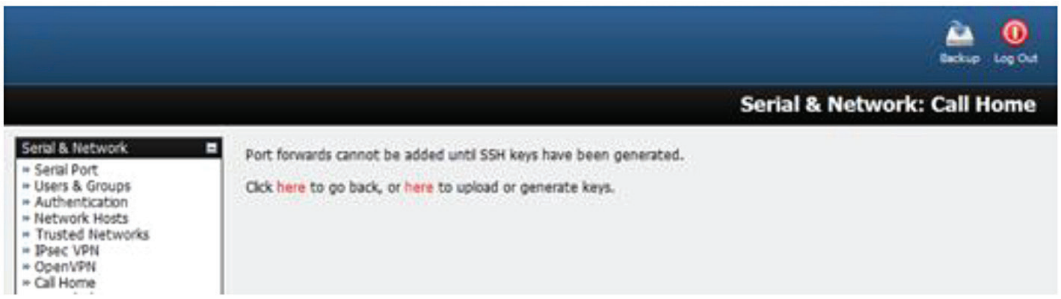


FIGURE 5-37. SERIAL AND NETWORK: CALL HOME SCREEN

- If you have not already generated or uploaded an SSH key pair for this console server, you will need to do so before proceeding (see Chapter 3).
- Click Add.

FIGURE 5-38. EDIT CONNECTION SCREEN

- Enter the IP address or DNS name (for example, the dynamic DNS address) of the VCMS.
- Enter the Password that you configured on the VCMS as the Call Home Password.
- Click Apply.

FIGURE 5-39. CALL HOME CONNECTION

This initiates the Call Home connection from the console server to the VCMS, creating an SSH listening port on the VCMS, and setting the console server up as a candidate.

Once the candidate has been accepted, an SSH tunnel to the console server is redirected back across the Call Home connection. The console server becomes a Managed Console Server and the VCMS can connect to and monitor it through this tunnel.

5.12.2 ACCEPT CALL HOME CANDIDATES AS MANAGED CONSOLES

This section gives an overview on configuring a VCMS to monitor console servers that Call Home. For more details, refer to the Virtual Central Management System (VCMS) User Manual.

NOTE For a VCMS to be contacted by the console server, it must have a static IP address or, if using DHCP, use a dynamic DNS service.

Enter a Call Home Password. This is used to accept connections from candidate console servers.

The Configure > Managed Console Servers screen on the VCMS shows the status of local and remote Managed Console Servers and candidates.

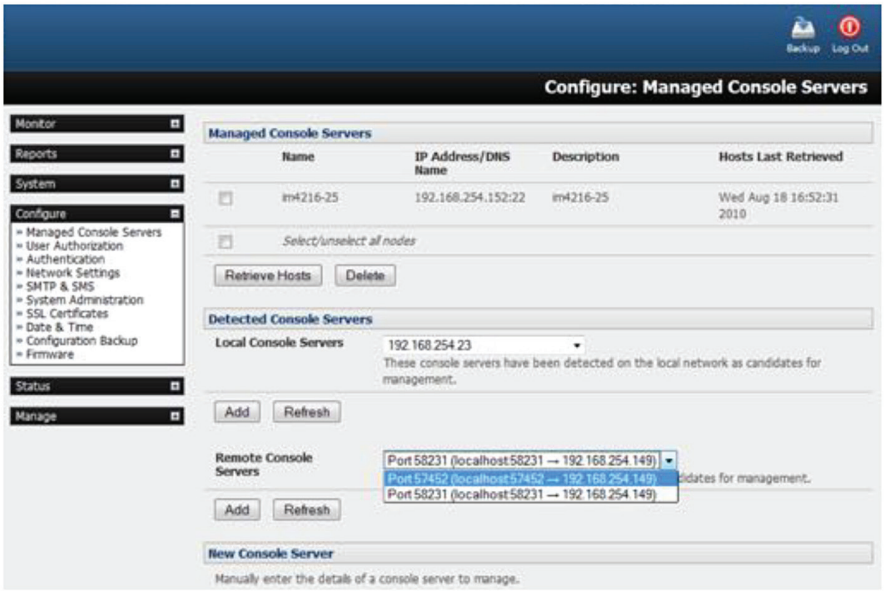


FIGURE 5-40. CONFIGURE; MANAGED CONSOLE SERVERS SCREEN

The Managed Console Server section shows the console servers currently being monitored by the VCMS.

The Detected Console Servers section shows the Local Console Servers drop down list (which lists all the console servers which are on the same subnet as the VCMS but are not currently being monitored) and the Remote Console Servers drop down list (which lists all the console servers that have established a Call Home connection but are not currently being monitored). Put another way, the Remote Console Servers drop-down list lists VCMS candidates.

To update either list, click Refresh.

To add a console server candidate to the Managed Console Server list:

- Select it from the Remote Console Servers drop down list.
- Click Add.
- Enter the IP Address and SSH Port (if these fields have not been auto-completed).
- Enter a Description and unique Name for the Managed Console Server you are adding.
- Enter the Remote Root Password (that is, the System Password that has been set on this Managed Console Server).

  This password is used by the VCMS to propagate auto generated SSH keys and then forgotten. It will not be stored.
- Click Apply.

The VCMS will now set up secure SSH connections to and from the Managed Console Server and will retrieve its Managed Devices, user account details and configured alerts.

5.12.3 CALLING HOME TO A GENERIC CENTRAL SSH SERVER

If you are connecting to a generic SSH server (not a VCMS) you may configure Advanced settings.

- Enter the SSH Server Port and SSH User to authenticate as.
- Enter the details for the SSH port forward(s) to create.

By selecting Listening Server, you may create a Remote port forward from the Server to this unit, or a Local port forward from this unit to the Server.

- Specify a Listening Port to forward from.

  Leave this field blank to allocate an unused port.
- Enter the Target Server and Target Port that will be the recipient of forwarded connections.
- Click Add.

5.13 IP PASSTHROUGH

IP Passthrough is used to make a modem connection (for example, the Black Box's internal cellular modem) appear like a regular Ethernet connection to a third-party downstream router, allowing the downstream router to use the Black Box's modem connection as a primary or backup WAN interface.

The Black Box provides the modem IP address and DNS details to the downstream device over DHCP and transparently passes network traffic to and from the modem and router.

While IP Passthrough essentially turns an Black Box into a modem-to-Ethernet half bridge, some specific layer 4 services (HTTP/HTTPS/SSH) may still be terminated at the Black Box (Service Intercepts). Also, services running on the Black Box can initiate outbound cellular connections independent of the downstream router.

This allows the console server to continue to be used for out-of-band management and alerting and also be managed via VCMS while in IP Passthrough mode.

5.13.1 DOWNSTREAM ROUTER SETUP

To use failover connectivity on the downstream router (aka Failover to Cellular or F2C), it must have two or more WAN interfaces.

NOTE: Failover in IP Passthrough context is performed entirely by the downstream router, and the built-in out-of-band failover logic on the console server itself is not available while in IP Passthrough mode.

- Connect an Ethernet WAN interface on the downstream router to the Console Server's Network Interface or Management LAN port with an Ethernet cable.
- Configure this interface on the downstream router to receive its network settings via DHCP.
- If failover is required, configure the downstream router for failover between its primary interface and the Ethernet port connected to the Console Server.

5.13.2 IP PASSTHROUGH PRE-REQUISITE PRE-CONFIGURATION STEPS

Configure the Network Interface and, where applicable, Management LAN interfaces with static network settings.

- Click Serial & Network > IP.
- For Network Interface and, where applicable, Management LAN, select Static for the Configuration Method and enter the network settings (see Section 4.3 for detailed instructions).
- For the interface connected to the downstream router, you may choose any dedicated private network. This network will only exist between the console server and downstream router and will not normally be accessible.
- For the other interfaces, configure as you would normally on the local network.
- For both interfaces, leave Gateway blank.

Configure the modem in Always On Out-of-band mode.

- For a cellular connection, click System > Dial > Internal Cellular Modem.
- Select Enable Dial-Out and enter carrier details such as APN (see Section 6.6 for detailed instructions).

5.13.3 IP PASSTHROUGH CERTIFICATION

To configure IP Passthrough:

- Click Serial & Network > IP Passthrough.

FIGURE 5-41. IP PASSTHROUGH SCREEN

- Check Enable.
- Select the modem to use for upstream connectivity.
- Optionally, enter the MAC Address of the downstream router's connected interface.

NOTE: If a MAC address is not specified, the console server will passthrough to the first downstream device requesting a DHCP address.

- Select the Ethernet Interface to use for connectivity to the downstream router.
- Click Apply.

5.13.4 SERVICE INTERCEPTS

These allow the console server to continue to provide services for out-of-band management when in IP Passthrough mode. Connections to the modem address on the specified intercept port(s) will be handled by the console server, rather than being passed through to the downstream router.

- For the required service of HTTP, HTTPS or SSH, check Enable.
- Optionally, modify the Intercept Port to an alternate port (for example, 8443 for HTTPS).

Do this if you want the downstream router to remain accessible via its regular port.

5.13.5 IP PASSTHROUGH STATUS

- Refresh the page to view the Status section.

  It displays the modem's External IP Address being passed through, the Internal MAC Address of the downstream router (only populated when the downstream router accepts the DHCP lease), and the overall running status of the IP Passthrough service.

Additionally, you may be alerted to the failover status of the downstream router by configuring a Routed Data Usage Check under Alerts & Logging > Auto-Response.

5.13.6 CAVEATS

Some downstream routers may be incompatible with the gateway route. This may happen when IP Passthrough is bridging a 3G cellular network where the gateway address is a point-to-point destination address and no subnet information is available.

The console server sends a DHCP netmask of 255.255.255.255. Most devices read this as a single host route. As an unusual Ethernet setting, older devices may have issues.

Intercepts for local services will not work if the console server is using a default route other than the modem. As per normal operation, they will also not work unless the service is enabled and access to the service is enabled (see System > Services > Service Access > Dialout/Cellular).

Outbound connections originating from console servers to remote services are supported (for example, sending SMTP email alerts, SNMP traps, getting NTP time, and IPSec tunnels). There is, however, a small risk of connection failure if both the console server and the downstream device try to access the same UDP or TCP port on the same remote host at the same time where they have randomly chosen the same originating local port number.

CHAPTER 6: FIREWALL, FAILOVER AND OOB ACCESS

The console server has a number of out-of-band access capabilities and transparent fail-over features, to ensure high availability. So if there's difficulty in accessing the console server through the main network path, all console server models provide out-of-band (OOB) access and the Administrator can still access it (and its Managed Devices) from a remote location.

All console server models support serially attaching an external dial-up modem and configuring dial-in OOB access. Some models with USB ports support attaching an external USB modem. Some models also come standard with an internal modem. These modems can also be configured for dial-in OOB access.

All console server models with an internal or externally attached modem (and V3.4 firmware or later) can be configured for out-dial to be permanently connected .

The advanced console server models can also be configured for transparent out-dial failover. So if the principal management network is disrupted, an external dial-up ppp connection is automatically established.

These advanced console server models can also be accessed out-of-band using an alternate broadband link and also offer transparent broadband failover.

Models with an internal cellular modem can be configured for OOB cellular access or for cellular transparent failover or can be configured as a cellular router.

6.1 DIAL-UP MODEM CONNECTION

To enable dial-in or dial-out you must first ensure there is a modem attached to the console server.

All LES1700-R2 come with an internal modem that can provide for OOB dial-in access. These models will display an Internal Modem Port tab under System > Dial as well as the Serial DB9 Port tab.

The LES1516A, LES1532A, LES1548A, and LES1600 models also support external USB modems. The USB modem will be auto-detected and an External USB Modem Port tab will come up under System > Dial in addition to the Serial DB9 Port tab. All console server models support an external modem (any brand) attached via a serial cable to the console/modem port for OOB dial-in access.

The serial ports on the LES1600 are, by default, all configured as RJ serial Console Server ports. Port 1 can be configured to be the Local Console/Modem port.

6.2 OOB DIAL-IN ACCESS

Once a modem has been attached to the console server, you can configure the console server for dial-in PPP access. The console server will then await an incoming connection from a dial-in at remote site. Next, the remote client dial-in software needs to be configured to establish the connection between the Administrator's client modem to the dial in modem on the console server.

6.2.1 CONFIGURE DIAL-IN PPP

Enable PPP access on the internal or externally attached modem:

- Navigate to System > Dial.

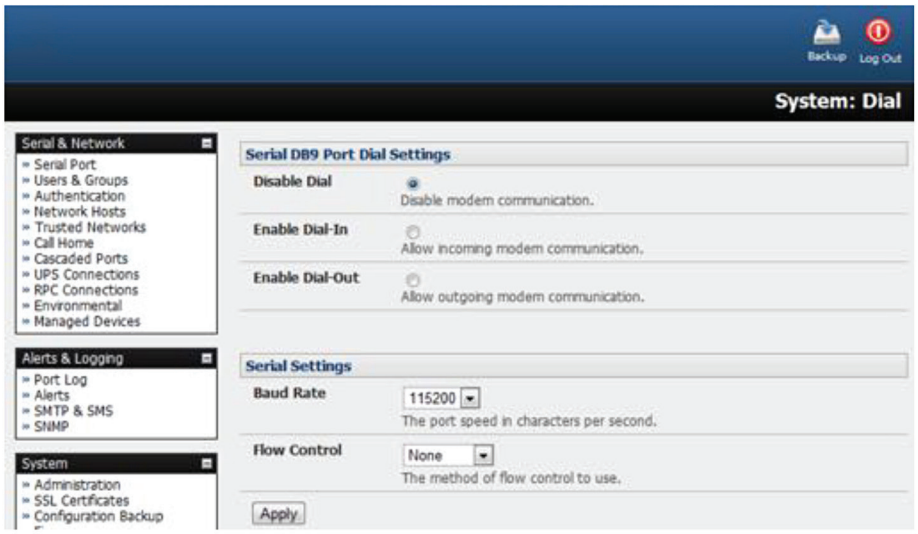


FIGURE 6-1. SYSTEM: DIAL SCREEN

- In the section appropriate to the port being configured (Serial DB9 Port or Internal Modem Port or External USB Port), select the Baud Rate and Flow Control that will communicate with the modem.

  By default, the modem port on all console servers is set with software flow control and the baud rate is set at:

  115200 baud for external modems connected to the local console port on LES1516A, LES1532A, LES1548A, and LES1700-R2 console servers.

  9600 baud for the internal modem or external USB modem and for external modems connected to the Console serial ports which have been reassigned for dial-in access (on LES1600).

  When enabling OOB dial-in we recommend that you change the Serial Settings to 38400 Baud Rate with Hardware Flow Control.

NOTE: You can further configure the console/modem port (e.g. to include modem init strings) by editing /etc/mgetty.config files as described in Chapter 15.

- Check the Enable Dial-In Access check box.
- In the Remote Address field, enter the IP address to be assigned to the dial-in client.

  You can select any address for the Remote IP Address. It must be in the same network range as the Local IP Address (for example, 200.100.1.12 and 200.100.1.67).

- In the Local Address field, enter the IP address for the Dial-In PPP Server.

  This is the IP address that will be used by the remote client to access console server once the modem connection is established. Again, you can select any address for the Local IP Address but it must both be in the same network range as the Remote IP Address.

- The Default Route option sets the dialed PPP connection as the default console server route.
- The Custom Modem Initialization option allows a custom AT string modem initialization string to be entered (e.g. AT&C1&D3&K3).

- Select the Authentication Type required.

Access is denied to remote users attempting to connect using an authentication scheme weaker than the selected scheme. The schemes, from strongest to weakest, are:

- Encrypted Authentication (MS-CHAP v2). Recommended. The strongest authentication.
- Weakly Encrypted Authentication (CHAP). This is the weakest encrypted password authentication to use. Not recommended as it provides very little password protection. Also note that clients connecting using CHAP are unable to encrypt traffic.
- Unencrypted Authentication (PAP). This is plain text password authentication. When using this type of authentication, the client password is transmitted unencrypted.
- None. No encryption at all.

- Select the Required Encryption Level. Access is denied to remote users attempting to connect not using this encryption level. 40 bit or 128 bit encryption is recommended.

NOTE: Firmware v3.5.2 and later support multiple dial-in users, setup with dialin Group membership. The User name and Password for the dial-in PPP link, and any dial-back phone numbers are configured during User set up. Earlier firmware only supports one PPP dial-in account.

Chapter 16 has Linux command examples to control modem port operation at the shell.

6.2.2 USING SDT CONNECTOR CLIENT

Administrators can use their SDT Connector client to set up secure OOB dial-in access to remote console servers. The SDT Connector Java client software provides point-and-click secure remote access. OOB access uses an alternate path for connecting to the console server to that used for regular data traffic.

Start an OOB connection in SDT Connector by initiating a dial-up connection, or adding an alternate route to the console server. SDT Connector allows for maximum flexibility in this regard, by allowing you to provide your own scripts or commands for starting and stopping the OOB connection. See Section 7.5 for more information.

6.2.3 SET UP WINDOWS XP OR LATER CLIENT

- Navigate to Start Menu > Control Panel.
- Click Network Connections.
- Click the New Connection Wizard.
- Select Connect to the Internet
- Click Next.
- On the Getting Ready screen, select Set up my connection manually.
- Click Next.
- On the Internet Connection screen select Connect using a dial-up modem.
- Click Next.
- Enter a Connection Name (any name you choose)
- Enter the dial-up Phone number that will connect thru to the console server modem.
- Enter the PPP User name and Password you have set up for the console server.

6.2.4 SET UP EARLIER WINDOWS CLIENTS

For Windows 2000, the PPP client set up procedure is the same as above, except you get to the Dial-Up Networking Folder by clicking Start and selecting Settings. Then click through Network > Dial-up Connections > Make New Connection.

6.2.5 SET UP LINUX CLIENTS

The online tutorial http://yolinux.com/TUTORIALS/LinuxTutorialPPP.html presents a selection of methods for establishing a dial up PPP connection.

- Command line PPP and manual configuration (which works with any Linux distribution).
- Using the Linuxconf configuration tool (for Red Hat compatible distributions).

  This configures the scripts ifup and ifdown to start and stop a PPP connection.
- Using the Gnome control panel configuration tool.
- Using WVDIAL and the Redhat Dialup configuration tool.
- Using the GUI dial program X-isp.

NOTE: For all PPP clients, set up TCP/IP as the only protocol enabled; set the Server to assign IP address and do DNS; do not set console server PPP as the default Internet connection.

6.3 DIAL-OUT ACCESS

A console server modem, internal or external, can be set in Failover mode (dialing-out after a ping failure) or with always-on dial-out. In either case, if disrupted, the console server tries to re-establish connection.

6.3.1 ALWAYS-ON DIAL-OUT

With firmware v3.4 and later console server modems can be configured for always-on dial-out, with a permanent external dial-up ppp connection.

- Navigate to System > Dial.

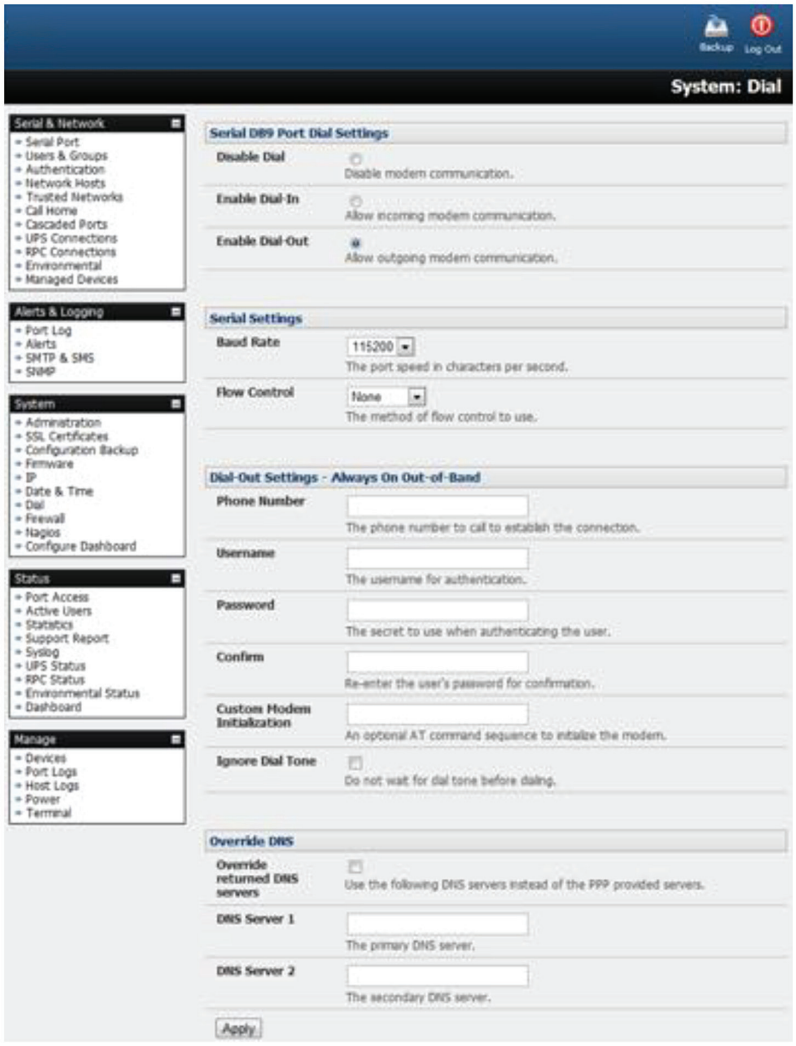


FIGURE 6-2. SYSTEM: DIAL SCREEN

- Check the Enable Dial-Out to allow outgoing modem communications.
- Select the Baud Rate and Flow Control that will communicate with the modem.
- In the Dial-Out Settings — Always On Out-of-Band fields enter the access details for the remote PPP server to be called.

The Override DNS section is available for PPP Devices such as modems. Override DNS allows the use of alternate DNS servers from those provided by your ISP. For example, an alternative DNS may be required for OpenDNS used for content filtering.

To enable Override DNS:

- Check the Override returned DNS Servers checkbox.
- Enter the IP address of the alternative DNS servers in the DNS Server 1 and DNS Server 2 entry fields.
- Click Apply.

6.3.2 FAILOVER DIAL-OUT

The LES1600, LES1516A, LES1532A, LES1548A, and LES1700-R2 series of advanced console servers can be configured so a dial-out PPP connection is automatically set up in the event of a disruption in the principal management network.

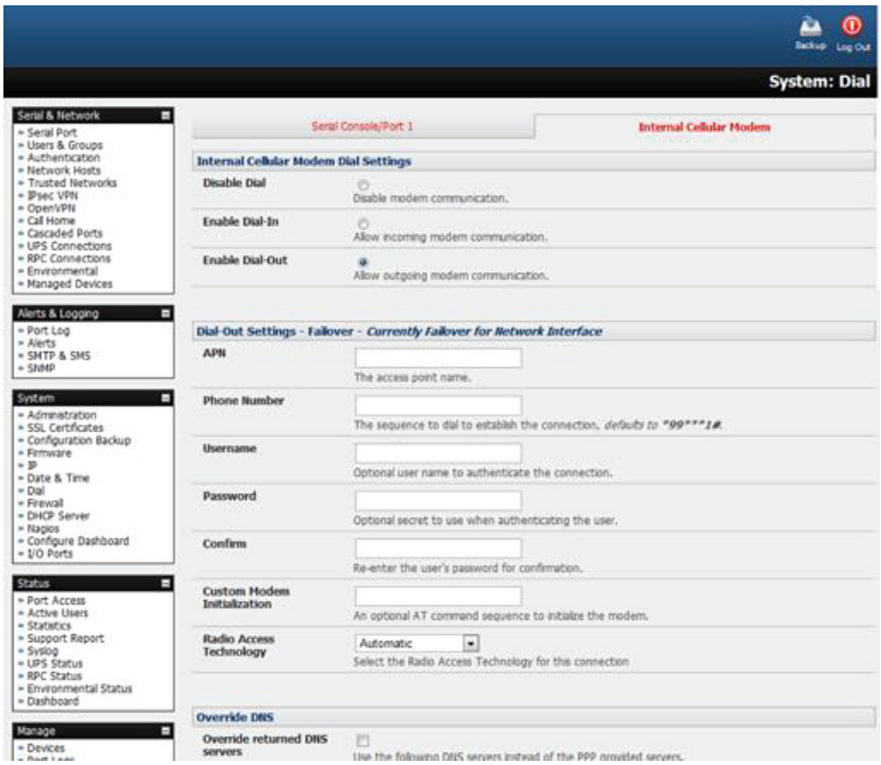


FIGURE 6-3.SET UP DIAL-OUT PPP CONNECTION SCREEN

NOTE: With firmware v3.0.1 and earlier, only SSH access is enabled on the failover connection. In firmware versions 3.0.2 and later, HTTPS access is also enabled. Once the dial-out PPP connection is established the administrator can connect to the console server via SSH (or HTTPS on console servers running firmware 3.0.2 or later) and fix the problem.

When configuring the principal network connection in System > IP, specify the Failover Interface to be used when a fault has been detected with Network or Network1 (that is, eth0). This can be either the Internal Modem, Dial Serial DB9 (if you are using an external modem on the console port), or USB Modem (if you are using a plug-on USB modem on an LES1600).

- Set the Probe Addresses of two sites (the Primary and Secondary) that the IM console server is to ping to determine if Network or Network1 is still operational.
- Navigate to System > Dial.
- Select the port to be configured: Serial DB9 Port, PC Card, or Internal Modem Port.

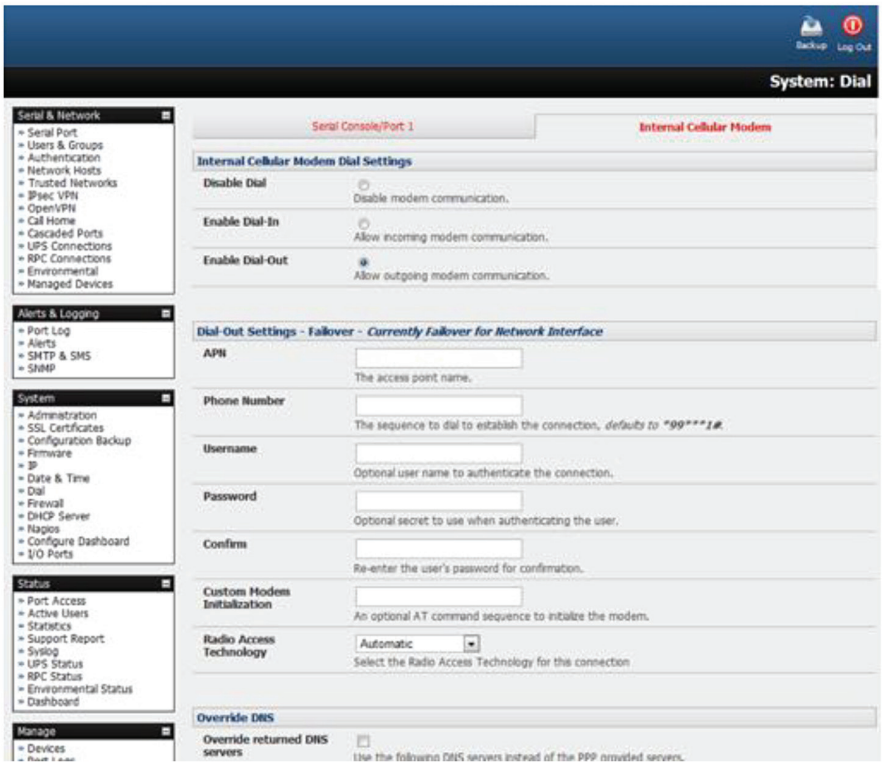


FIGURE 6-4.

- Select the Baud Rate and Flow Control that will communicate with the modem.
- Check the Enable Dial-Out Access checkbox.
- In the Dial-Out Settings — Always On Out-of-Band fields enter the access details for the remote PPP server to be called.

The Override DNS section is available for PPP Devices such as modems. Override DNS allows the use of alternate DNS servers from those provided by your ISP. For example, an alternative DNS may be required for OpenDNS used for content filtering.

To enable Override DNS:

- Check the Override returned DNS Servers checkbox.
- Enter the IP address of the alternative DNS servers in the DNS Server 1 and DNS Server 2 entry fields.
- Click Apply.

NOTE: As of firmware v3.1.0 and later, the advanced console server, by default, supports automatic failure-recovery back to the state prior to the failover. The advanced console server continually pings probe addresses while in original and failover states. The original state will automatically be set as a priority and reestablished following three successful pings of the probe addresses during failover. The failover state will be removed once the original state has been re-established.

6.4 OOB BROADBAND ETHERNET ACCESS

The LES1600, LES1516A, LES1532A, LES1548A, and LES1700-R2 family of advanced console servers have a second ethernet port that can be configured for alternate and OOB (out-of-band) broadband access.

TABLE 6-1. SECOND ETHERNET PORT TO CONFIGURE FOR OOB BROADBAND ACCESS

PRODUCT CODE	LABEL INDICATING SECOND ETHERNET PORT
LES1516A, LES1532A, LES1548A and LES1600	NET2
LES1700-R2 series	NET2

With two active broadband access paths to these advanced console servers, in the event you are unable to access through the primary management network (LAN1, Network or Network1) you can still access it through the alternate broadband path.

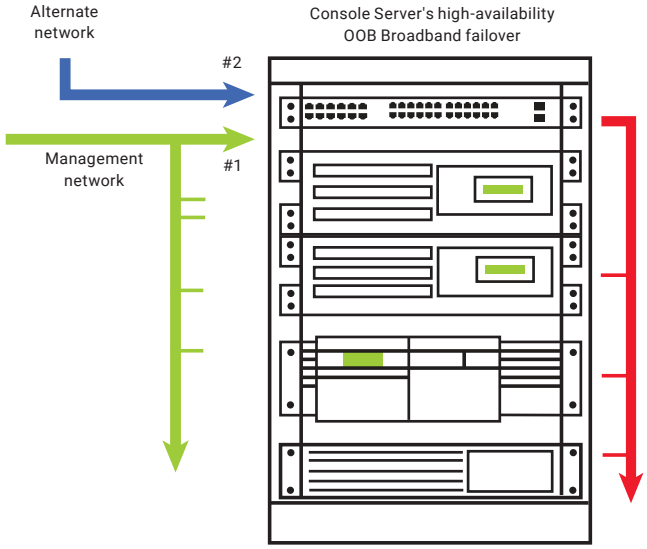


FIGURE 6-5. OOB BROADBAND FAILOVER

- Navigate to System > IP.
- Select Management LAN Interface (LES1516A, LES1532A, LES1548A, and LES1700-R2).
- Configure the IP Address, Subnet Mask, Gateway and DNS with the access settings that relate to the alternate link.

NOTE: When configuring the principal Network Interface connection, the Failover Interface must be set to None.

6.5 BROADBAND ETHERNET FAILOVER

The second Ethernet port on the LES1600, LES1516A, LES1532A, LES1548A, and LES1700-R2 family of advanced console servers can also be configured for failover to ensure transparent high availability.

- Navigate to System > IP > Network Interface.

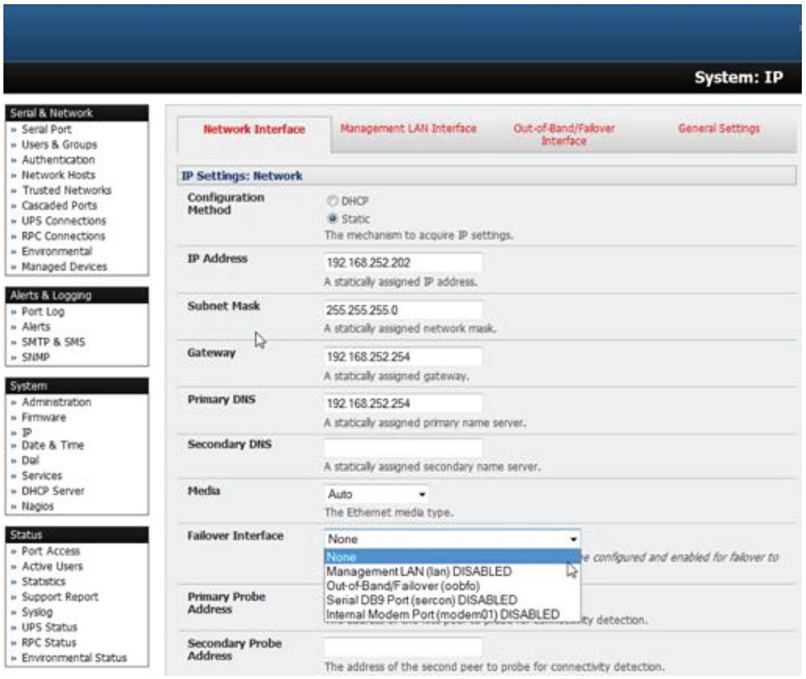


FIGURE 6-6. SYSTEM IP: NETWORK INTERFACE TAB

- Select Management LAN from the Failover Interface pop-up menu.
- Enter the Primary Probe Address and the Secondary Probe Address.

  These are the IP addresses or hostnames of the two hosts (the Primary and Secondary) that the advanced console server is to ping to determine if a Network Interface is still operational.
- Select the Out-of-Band/Failover Interface tab.
- Enter the Out-of-Band/Failover IP Address, Subnet Mask, and Gateway values.

  These values should be the same as used for the Network Interface.

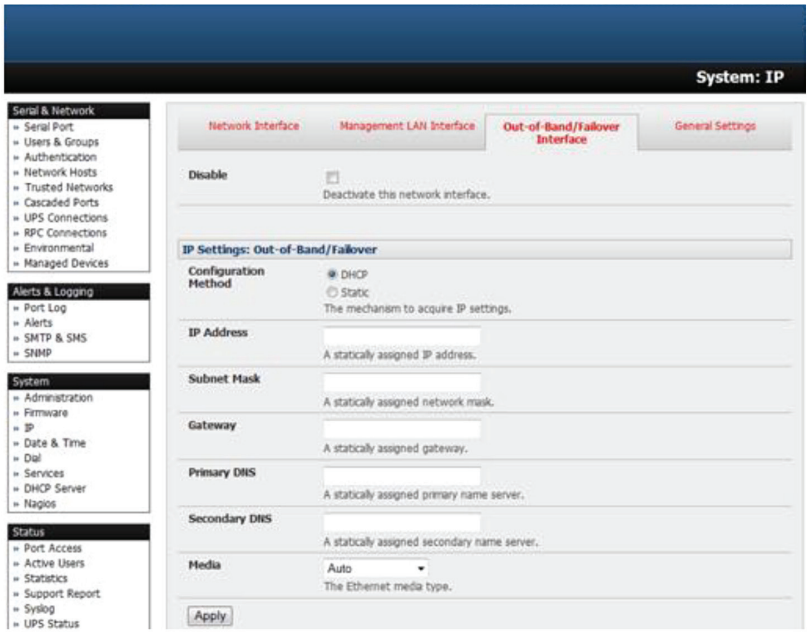


FIGURE 6-7. SYSTEM IP: FAILOVER INTERFACE TAB

In this mode, the Management LAN Interface is available as the transparent back-up port to Network Interface for accessing the management network. Management LAN Interface will automatically and transparently take over the work of Network Interface, if the Network Interface becomes unavailable for any reason.

NOTE: In console servers running firmware v3.0.1 and earlier, only SSH access is enabled on the failover connection. In firmware versions 3.0.2 and later, HTTPS access is also enabled. Once the dial-out PPP connection is established, the administrator can connect to the console server via SSH (or HTTPS on console servers running firmware 3.0.2 or later) and fix the problem.

As of firmware v3.1.0 and later, the advanced console server, by default, supports automatic failure-recovery back to the state prior to the failover. The advanced console server continually pings probe addresses while in original and failover states. The original state will automatically be set as a priority and re-established following three successful pings of the probe addresses during failover. The failover state will be removed once the original state has been re-established.

For firmware versions prior to v3.1.0 the advanced console server does not support automatic failure-recovery back to the original state prior to the failover. To restore networking to a recovered state, the following command then needs to be run:

```
rm -f /var/run/*-failed-over && config -r ipconfig
```

If required, you can run a custom bash script when the device fails over. It is possible to use this script to implement automatic failure recovery, depending on your network setup. The script to create is:

```
/etc/config/scripts/interface-failover-alert
```

6.6 CELLULAR MODEM CONNECTION

The LES1600 family of advanced console servers support internal cellular modems.

These modems first need to be installed (as documented in Sections 6.6.1 through 6.6.3) and then set up to validate they can connect to the carrier network (as documented in Sections 6.6.4 and 6.6.5).

They then can be configured for operation in Always-on cellular router or OOB mode, or in Failover mode (as documented in Section 6.7).

6.6.1 CONNECTING TO A GSM HSUPA/UMTS CARRIER NETWORK

Console server models denoted with -R have an internal GSM modem that will connect to any major GSM carrier globally.

NOTE: Before powering on any -R model console server, install the SIM card provided by your cellular carrier and attach the external aerial. The console server has two cellular status LEDs. The SIM LED on top of the unit should go on solid when a SIM card has been inserted and detected.

- Navigate to System > Dial.
- Click the Internal Cellular Modem tab.

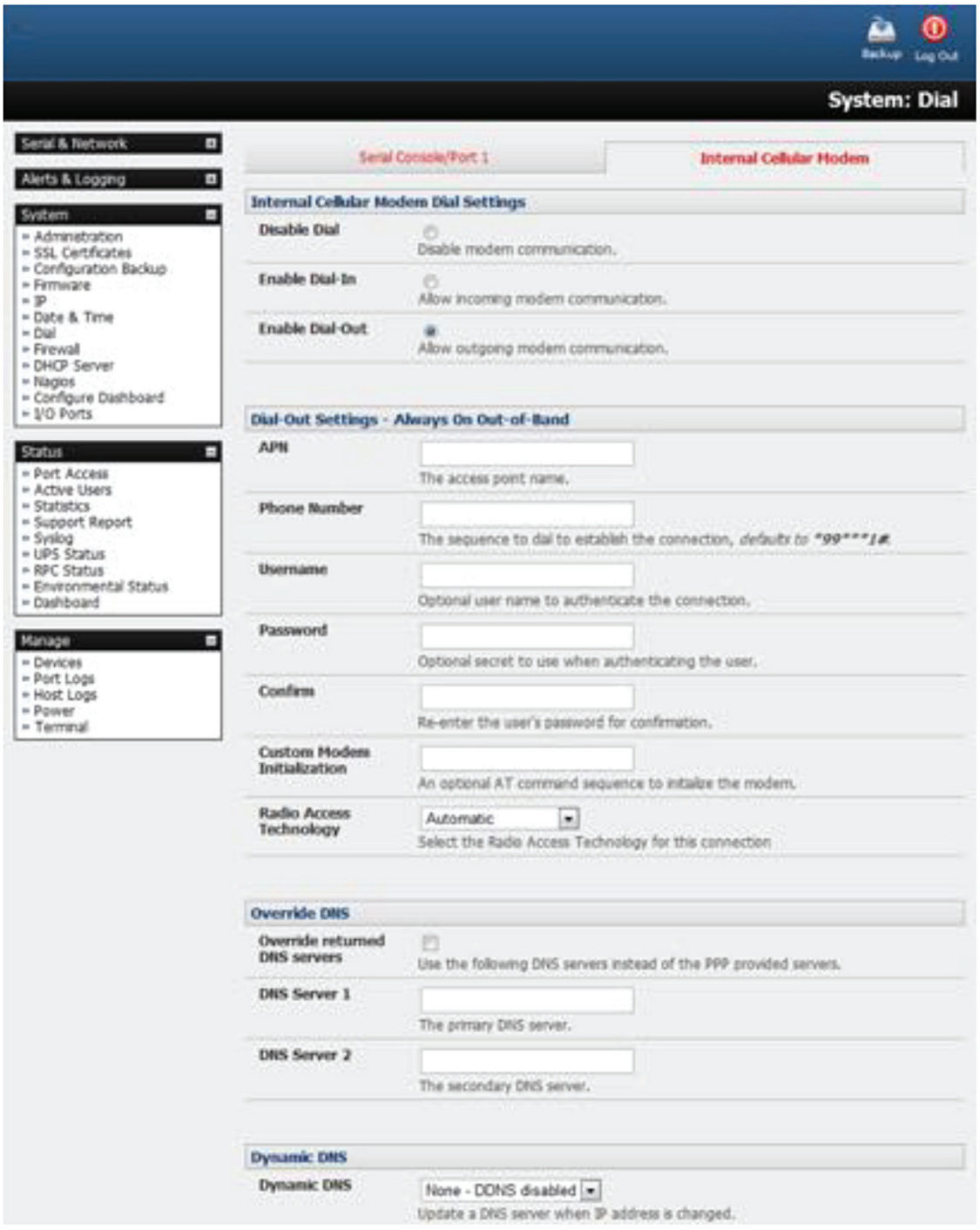


FIGURE 6-8. INTERNAL CELLULAR MODEM TAB

- Check the Enable Dial-Out radio button in the Internal Cellular Modem Dial Settings section.

Your carrier may have provided details for configuring the connection:

TABLE 6-2. CONFIGURATION DETAILS FROM CARRIER

VALUE	DESCRIPTION
APN	Access Point name
PIN Code	If the carrier-provided SIM card is locked, a PIN Code may be required to unlock it.
Phone Number	The dial sequence which establishes the connection. By default this is *99***1#.
Username	Optional
Password	Optional
Custom Modem Initialization	Optional AT command sequence to initialize the modem.

- Enter the carrier's APN.

Example APNs include:

TABLE 6-3. EXAMPLE APNS

CARRIER	APN
AT&T (USA)	i2gold
T-Mobile (USA)	epc.tmobile.com
Internode (Australia)	internode
Telstra (Australia)	telstra.internet

NOTE: The APN is, in most cases, the only value needed. The other fields can be left blank.

- If the SIM Card is configured with a PIN Code, unlock the Card by entering the PIN Code.

NOTE: If the PIN Code is entered incorrectly three times, the PUK Code will be required to unlock the Card.

You may also need to use Override DNS to set alternate DNS servers from those provided by your carrier. If this is necessary:

- On System > Dial > Internal Cellular Modem, check the Override returned DNS servers checkbox.
- Enter the alternative DNS servers in the DNS Server 1 and DNS Server 2 fields.
- Click Apply.

  A radio connection will be established with your cellular carrier.

6.6.2 CONNECTING TO A CDMA EV-DO CARRIER NETWORK

Console server models denoted with -V have an internal CDMA modem and will connect to the Verizon network in North America.

After creating an account with the CDMA carrier, some carriers require an additional step to provision the Internal Cellular Modem, known as Provisioning. The console servers support:

- Over-the-Air Service Provisioning (OTASP) where modem-specific parameters can be retrieved via a voice call to a special phone number, and
- a manual process where the phone number and other parameters are entered manually.

OTASP

NOTE Before this can be achieved, a working account and an activated device are required. In this case, an activated device is a Black Box console server which has had its ESN (Electronic Serial Number) registered with an appropriate plan on your carrier's account.

- Navigate to System > Dial.
- Click the Internal Cellular Modem tab.
- Enter the particular phone number which must be dialed to complete OTASP.

For example, Verizon uses *22899 and Telus uses *22886.

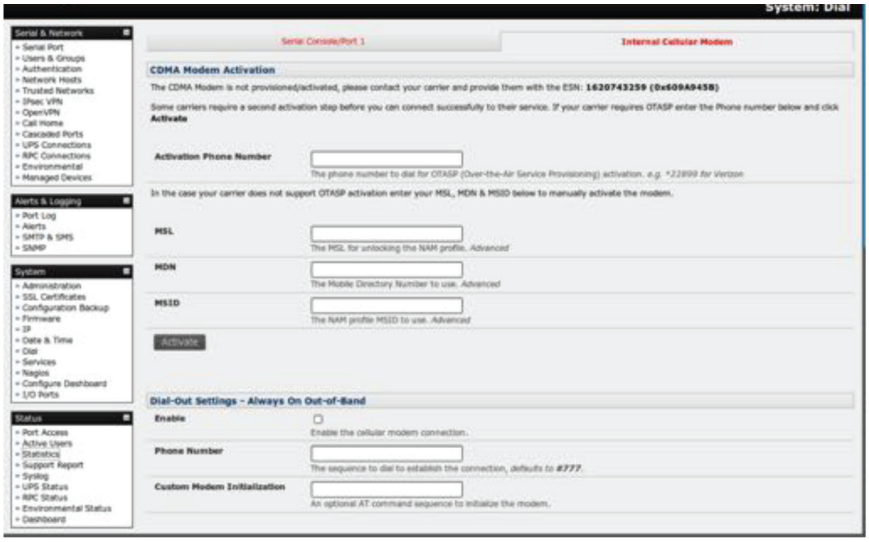


FIGURE 6-9.

- Click Activate. This initiates the OTASP call.

  The process is successful if no errors are displayed and you no longer see the CDMA Modem Activation form.

  If OTASP is unsuccessful consult the System Logs at Status > Syslog for clues to what went wrong.
- When OTASP has completed enable the Internal Cellular Modem by entering the carrier's phone number.

  By default, this number is #777.
- Click Apply.

To confirm OTASP success, and to display the modem's current state:

- Navigate to Status> Statistics.

- click the Cellular tab.

  The current state of the modem will present.

OTASP success will result in a valid phone number being placed in the NAM Profile Account MDN field.

Manual Activation

If a carrier does not support OTASP, you will need to manually provision the modem.

- Navigate to System > Dial.
- Click the Internal Cellular Modem tab.

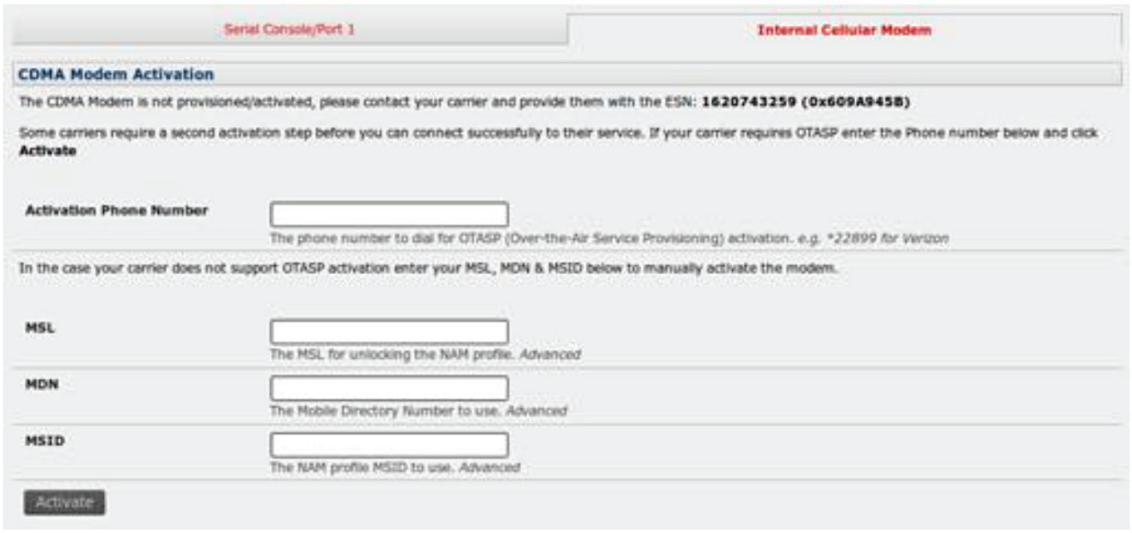


FIGURE 6-10. INTERNAL CELLULAR MODEM TAB

- Enter the MSL, MDN and MSID values.

  These values are specific to your carrier and for manual activation, you will have to learn what values your carrier uses in each field.

  Verizon, for example, has been known to use an MSL of 000000 and the phone number assigned to the Black Box device as both the MDN and MSID with no spaces or hyphens. An assigned phone number of 555-123-1234 is entered in the MDN and MSID fields as 5551231234.
- Click Activate.

If no errors occur you will see the new values entered into the NAM Profile Account. To check this:

- Navigate to Status> Statistics.
- Click the Cellular tab.

To connect to your carrier's 3G network:

- Navigate to System > Dial.
- Click the Internal Cellular Modem tab.
- Enter the appropriate Phone Number.

  This is usually #777.

- If required by your account plan, enter the supplied Username and Password.
- Check the Enable check-box.
- Click Apply.

  The Always On Out-of-Band connection is initiated.

6.6.3 CONNECTING TO A 4G LTE CARRIER NETWORK

Console server models denoted with -V, -T, or -R have an internal modem that will connect to any major 4G LTE carrier globally.

NOTE: Before powering on any -V, -T, or -R model console server, install the SIM card provided by your cellular carrier and attach the external aerial.

- Navigate to System > Dial.
- Click the Internal Cellular Modem tab.
- Check the Enable Dial-Out radio button in the Internal Cellular Modem Dial Settings section.

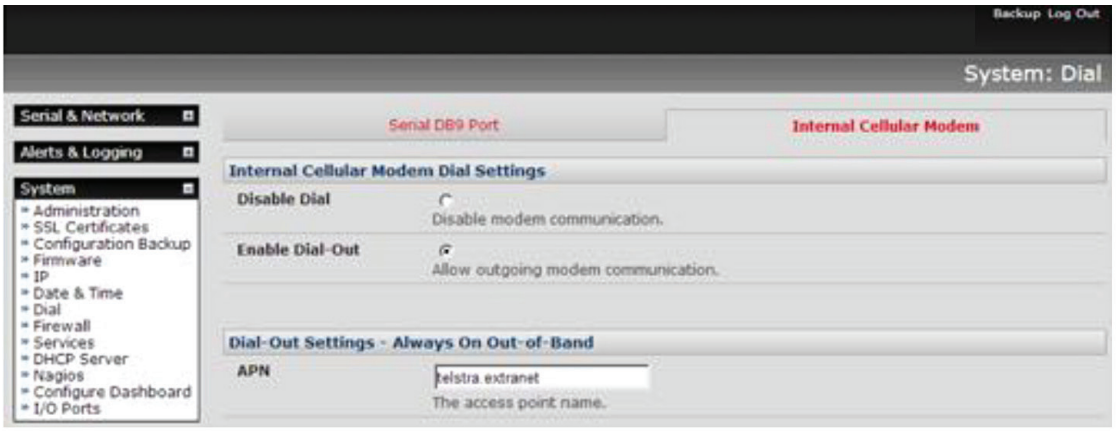


FIGURE 6-11. INTERNAL CELLULAR MODEM TAB

Your carrier may have provided details for configuring the connection.

TABLE 6-4. CARRIER-PROVIDED DETAILS

VALUE	DESCRIPTION
APN	Access Point Name
PIN Code	If the carrier-provided SIM card is locked, a PIN Code is required to unlock it.
Phone number	The dial sequence to establish the connection. By default this is *99***1#.
Username	Optional
Password	Optional

- Enter the carrier's APN.

Example APNs include:

TABLE 6-5. EXAMPLE APNS

CARRIER	APN
AT&T (USA)	i2gold
T-Mobile (USA)	epc.tmobile.com
Internode (Australia)	internode
Telstra (Australia)	telstra.internet

NOTE: The APN is, in most cases, the only value needed. The other fields can be left blank.

- If the SIM Card is configured with a PIN Code, unlock the Card by entering the PIN Code.

NOTE: If the PIN Code is entered incorrectly three times, the PUK Code will be required to unlock the Card.

You may also need to use Override DNS to set alternate DNS servers from those provided by your carrier. If this is necessary:

- On System > Dial > Internal Cellular Modem, check the Override returned DNS servers checkbox.

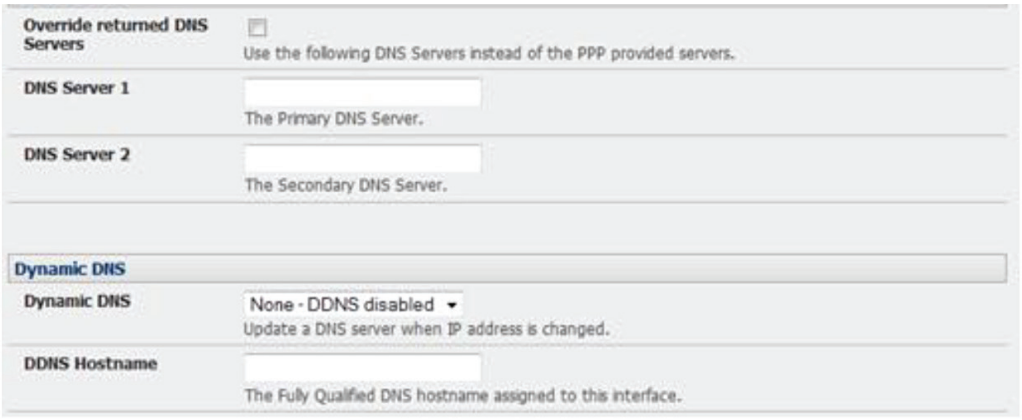


FIGURE 6-12. OVERRIDE RETURNED DNS SERVERS

- Enter the alternative DNS servers in the DNS Server 1 and DNS Server 2 fields.
- Click Apply.

  A radio connection will be established with your cellular carrier.

6.6.4 VERIFYING THE CELLULAR CONNECTION

Out-of-band access is enabled by default so the cellular modem connection should now be on. To verify this:

- Navigate to Status > Statistics.
- Select the Cellular tab.
- Verify the Mode is set to Online.
- Select the Failover & Out-of-Band tab.

- Verify the Connection Status reads as Connected.

To measure the received signal strength:

- Navigate to Status > Statistics.

The current state of the cellular modem, including the Received Signal Strength Indicator (RSSI), will present. Note the RSSI coverage value.

TABLE 6-6. CURRENT STATE OF CELLULAR MODEM

RECEIVED SIGNAL STRENGH INDICATOR (RSSI) VALUE	DESCRIPTION
≤ -100 dBm)	unacceptable
-99 to -90 dBm	weak-to-medium
-89 to -70 dBm	medium-to-strong
≥ -69 dBm	very strong

RSSI is a measure of the Radio Frequency (RF) power present in a received radio signal. It is generally expressed in decibel-milliwatts (dBm). The best throughput comes from placing the receiving device in a location with the highest possible RSSI.

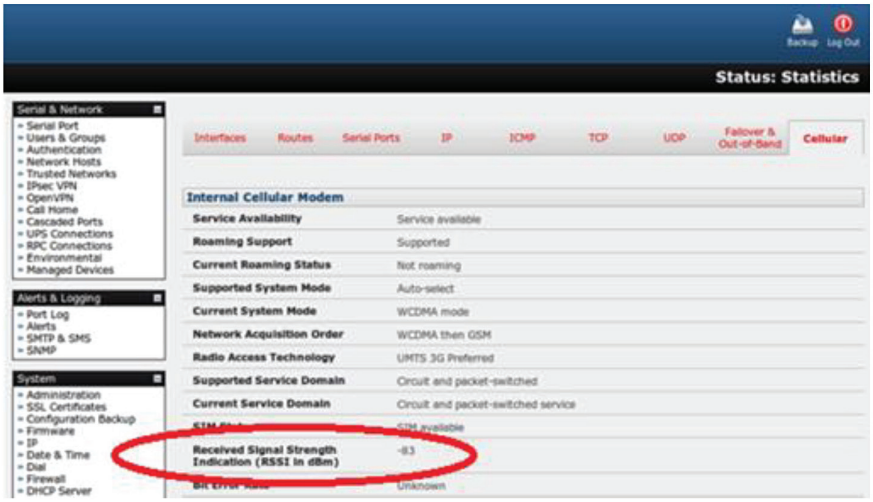


FIGURE 6-13. RSSI

With the cellular modem connection on, the connection status is also visible via the LEDs on top of console server.

NOTE: The LES1204A-G has two cellular status LEDs. The WWAN LED is OFF when in reset mode or not powered. When powered, it will go ON and while searching for service it will flash off briefly every five seconds.

6.6.5 CELLULAR MODEM WATCHDOG

As of with firmware V3.5.2u13 and later, when you check the Enable Dial-Out check-box at System > Dial, you will be given the option to configure a cellar modem watchdog service.

This service will periodically ping a configurable IP address. If a threshold number of consecutive attempts fail, the service will cause the unit to reboot. This can be used to force a clean restart of the modem and its services to work around any carrier issues.

6.6.6 DUAL-SIM FAILOVER

Some console server models allow you to insert two SIM cards, allowing for selective connection to two carrier networks. The dual-SIM failover feature allows the cell modem to selectively failover to the secondary SIM when communication over the primary SIM fails.

To configure dual-SIM failover:

- Navigate to System > Dial.
- Click the Internal Cellular Modem tab.
- In the SIM Configuration section, set the Primary SIM to either Bottom Slot or Top Slot.

Choose the slot which contains the SIM from your primary carrier network.

- Check the Enable SIM Failover checkbox.
- Specify how the device will failback from the failover SIM to the Primary SIM.

  There are two options:

  - On Disconnect. With this option the console server will failback to the Primary SIM only after the connection on the failover SIM has failed its ping test.
  - On Timeout. With this option, the console server will failback to the Primary SIM after the connection on the failover SIM has been up for the timeout period.
- If On Timeout is the selected failback option, set the Failback Timout value.
  - The Failback Timout is the number of seconds the failover SIM must be connected before the console server switches back to the Primary SIM.
  - If no number is entered here, the default value of 600 seconds (10 minutes) applies.
- Configure each SIM connection with the information necessary (APN, and, if required, the PIN, Phone Number, Username and Password) to enable it to make a successful connection, assuming sufficient signal strength from the cell service provider.

  See Sections 6.6.1 through 6.6.3 for details.
- Enter a Failback Test IP address for each SIM.

  This IP address is used to ping test the status of the cell modem connection and to determine if SIM failover or failback is to take place.
- Optionally configure DDNS and the Modem Watchdog (see Section 6.6.5).

  DDNS, when configured, will be applied to the cell modem dial out connection regardless of which SIM is currently in use.

NOTE: Dual-SIM failover is for dial-out connections only.

6.6.7 MULTI-CARRIER CELLULAR SUPPORT

Some cellular carriers require the console server's cellular modem to be programmed with carrier-specific firmware to operate on their network. Some console server models, however, are equipped with a reprogrammable cellular modem, allowing them to operate on more than one such carrier network.

NOTE: Changes to the cellular modem firmware are unaffected by Black Box firmware upgrades or factory erase/configuration reset operations.

On console servers with multi-carrier capability:

- Navigate to System > Dial.
- Select the Internal Cellular Modem tab.

FIGURE 6-14. INTERNAL CELLULAR MODEM TAB

The Internal Cellular Modem Carrier Settings section (which provides control over which carrier's firmware is installed on the modem) will present.

- Select the desired Carrier radio button.

  The modem's flash memory will have the carrier-specific firmware image installed.

  Flashing takes several minutes during which the cellular modem is unavailable. During this time, the page periodically refreshes with status information.

  Upon successful completion, the page displays the message: Cellular Firmware carrier change completed.

Multi-carrier capable models ship with cellular modem firmware for each supported carrier pre-loaded onto internal non-volatile or USB storage. Periodically, new cellular modem firmware becomes available and is published on the Black Box downloads site.

NOTE: If your unit's cellular connection is operating correctly, there is typically no need to upgrade its cellular firmware.

On console servers with multi-carrier capability, to download and apply new cellular firmware using the Management Console UI:

- Navigate to System > Firmware.

  A section presents showing the local cellular firmware image status and a Check for Update button that starts the firmware update process.

  The Cellular Firmware Status section indicates the date of the last firmware download, and shows a cryptographic fingerprint.

- Click the Check for Update button.

  The Management Console contacts the remote server, ftp://ftp.Black Box.com/, and displays an update summary.

  This summary indicates the local and remote fingerprints for comparison, without altering any local files.

- Optionally, expand the Advanced section.

  This section shows a full list of files to be downloaded or deleted, along with their SHA1 hashes. (Temporary files downloaded during the initial Check for Updates may be listed as simple files to copy into place, as they do not have to be re-downloaded.)

- Click Download and Apply.

NOTE: The modem will only be flashed if new firmware is available for the currently selected carrier.

During the download and flashing of the firmware, an interstitial screen displays, showing Currently upgrading cellular modem firmware. Once completed, the status of the firmware update is displayed at System > Firmware.

- Alternatively, click Cancel to reject the update.

You can also control multi-carrier features at the console server shell.

- Show currently selected carrier.

  cellctl -is | egrep "^preferred-carrier" | cut -d " " -f 2

- Show current modem firmware version.

  cellctl -is | egrep "^current-firmware" | cut -d " " -f 2

- List available carriers supported on the installed modem.

  /etc/scripts/cell-fw-update -l

- Check for availability of firmware updates.

  /etc/scripts/cell-fw-update -u

  Output is the remote fingerprint followed by the list of actions that would be taken by cell-fw-update -d.

- Download latest firmware for all carriers supported by the modem.

  /etc/scripts/cell-fw-update -d

- Flash modem with latest local firmware for carrier.

  /etc/scripts/cell-fw-update -c <carrier>

  <carrier> is one of the carrier identifiers emitted by cell-fw-update -l.

  This command can be used to switch carriers or to update the firmware of the current carrier.

NOTE: If the firmware version information on the modem is identical, the modem may reject the update without error.

6.7 CELLULAR OPERATION

When set up as a console server, the 3G cellular modem can be set up to connect to the carrier in one of four modes.

- OOB mode. In this mode, the dial-out connection to the carrier cellular network is always on, awaiting incoming access from a remote site wanting to access the console server or attached serial consoles/network hosts.
- Failover mode. In this mode, a dial-out cellular connection is only established in event of a ping failure.
- Cellular router mode. In this mode, the dial-out connection to the carrier cellular network is always on, and IP traffic is routed between the cellular connected network and the console server's local network ports.
- Circuit Switched Data (CSD) mode. In this dial-in mode, the cellular modem can receive incoming calls from remote modems who dial a special Data Terminating number. This is a 3G-only mode.

6.7.1 OOB ACCESS SETUP

In this mode, the dial-out connection to the carrier cellular network is always on, awaiting any incoming traffic. By default, the only traffic enabled is incoming SSH access to the console server and its serial ports, and incoming HTTPS access to the console server. There is a low level of keep alive and management traffic going over the cellular network. Generally, the status reports, alerts and other traffic from the site can be carried over the main network.

This mode is typically used for out of band access to remote sites. Consequently, to be directly accessed, the appliance needs to have a Public IP address and it must not have SSH access firewalled. This OOB mode is the default for LES1700-R2 and LES1400 appliances with internal cellular modems. Out-of-band access is enabled by default and the cellular modem connection is always on.

Almost all carriers offer corporate mobile data service/plans with a Public (static or dynamic) IP address. These plans often have a service fee attached.

With a static Public IP address plan, you can try accessing the console server using the Public IP Address provided by the carrier. By default, only HTTPS and SSH access is enabled on the OOB connection: you can browse to the console server, but you cannot ping it.

With a dynamic Public IP address plan, a DDNS service will need to be configured to allow the remote administrator to initiate incoming access. Once this is done, you can then also try accessing the console server using the allocated domain name.

By default, most providers offer a consumer-grade service that provides dynamic Private IP address assignments to 3G devices. This IP address is not visible across the Internet, but usually it is adequate for home and general business use.

To confirm a consumer-grade service:

- Navigate to the Status > Statistics.
- Click the Failover & Out-of-Band tab.
- In the Always on Out-of-Band — Internal Cellular Modem (cellmodem) section, check the value presented for IP Address.

FIGURE 6-15.

If the value is in one of the private IP Address ranges —

10.0.0.0 – 10.255.255.255

172.16.0.0 – 172.31.255.255

192.168.0.0 – 192.168.255.255

— you have a consumer-grade cellular service.

For inbound OOB connection with such a plan, you will need to use Call Home with a VCMS or set up a VPN.

In Out of Band access mode, the internal cellular modem will continually stay connected. The alternative is to set up Failover mode on the console server.

6.7.2 CELLULAR FAILOVER

In this mode, a dial-out cellular connection is only established if the main network is disrupted. The cellular connection normally remains idle and in a low power state. It is only activated if there is a ping failure. This standby mode suits remote sites with expensive power or high cellular traffic costs.

In this mode, the appliance continually pings nominated probe addresses over the main network connection. If there is a ping failure, it dials out and sets up a dial-out ppp connection over the cellular modem and access is switched transparently to this network connection. Then when the main network connection is restored, access is switched back.

Once you have configured the carrier connection, the cellular modem can be configured for failover.

This will tell the cellular connection to remain idle in a low power state. If the primary and secondary probe addresses are not available, it will bring up the cellular connection and connect back to the cellular carrier.

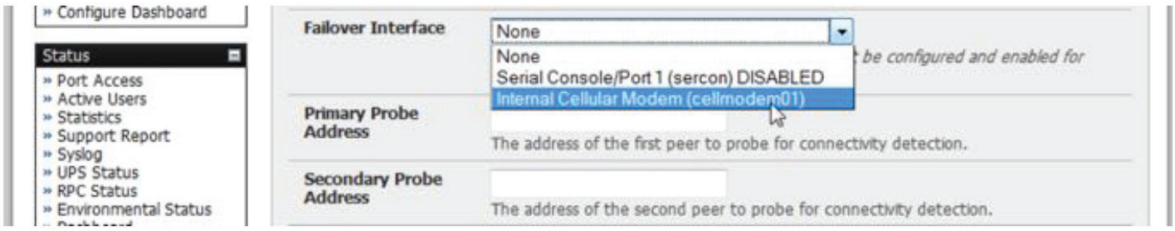


FIGURE 6-16.

- Navigate to System > IP.
- Select Internal Cellular Modem (cellmodem01) from the Failover Interface pop-up menu.
- Enter the Primary Probe Address and the Secondary Probe Address.

  These are the two sites the console server pings to determine if the principal network is operational.

If the principal network fails, the cellular network connection is activated as the access path to the console server and any managed devices.

NOTE: Only HTTPS and SSH access are enabled on the failover connection. This allows an administrator to connect to the console server to diagnose and correct the network failure without offering third-parties a large attack surface.

As of firmware v3.1.0, the advanced console server supports automatic failure-recovery back to the original state before failover by default.

The advanced console server continually pings probe addresses while in original and failover states. The original state will automatically be set as a priority and re-established following three successful pings of the probe addresses during failover. The failover state will be removed once the original state has been re-established.

For earlier firmware, which does not support automatic failure-recovery, to restore networking to a recovered state you need to run the following command:

```
rm -f /var/run/*-failed-over && config -r ipconfig
```

If required, you can run a custom bash script when the device fails over. You can use this script to implement automatic failure recovery, depending on your network setup. The script to create is:

```
/etc/config/scripts/interface-failover-alert
```

To check the connection status:

- Navigate to Status > Statistics.
- Click the Failover & Out-of-Band tab.

FIGURE 6-17.

- Note the Active Connection value.

  If the Main Connection is good, the Active Connection value will be Main.

  If the Main Connection is down, the Out-of-Band/Failover section displays information relating to a configured Out-of-Band/Failover interface and the status of that connection. The IP Address of the Out-of-Band/Failover interface will be presented in the Out-of-Band/Failover section once the Out-of-Band/Failover connection has been triggered and made.

6.7.3 CELLULAR ROUTING

Once you have a configured carrier connection, the cellular modem can be configured to route traffic through the console server. This requires setting up forwarding and masquerading as detailed in Section 6.8.

6.7.4 CELLULAR CSD DIAL-IN

CSD is a legacy form of data transmission developed for TDMA-based mobile phone systems like GSM. CSD uses a single radio time slot to deliver 9.6 kbps data transmission to the GSM Network and Switching Subsystem where it could be connected through the equivalent of a normal modem to the Public Switched Telephone Network (PSTN) allowing direct calls to any dial-up service.

CSD is provided selectively by carriers and it is important you receive a Data Terminating number as part of the mobile service your carrier provides. This is the number that external modems will call to access the console server.

Once you have configured carrier connection, the cellular modem can be configured to receive Circuit Switched Data (CSD) calls.

- Navigate to System > Dial.
- Click the Internal Cellular Modem tab.
- Check the Enable Dial-In radio button.
- Enter the required information in the Dial-In Settings section.

6.8 FIREWALLS AND FORWARDING

Console servers with firmware v3.3 and later have basic routing, NAT (Network Address Translation), packet filtering and port forwarding support on all network interfaces.

This enables the console server to function as an Internet or external network gateway, via cellular connections or via other Ethernet networks on two Ethernet port models.

Network Forwarding allows the network packets on one network interface (for example LAN1, aka eth0) to be forwarded to another network interface (for example, LAN2 or dial-out/cellular). Locally networked devices can IP connect through the console server to devices on remote networks.

IP Masquerading is used to allow all the devices on your local private network to hide behind and share the one public IP address when connecting to a public network. This type of translation is only used for connections originating within the private network destined for the outside public network, and each outbound connection is maintained by using a different source IP port number.

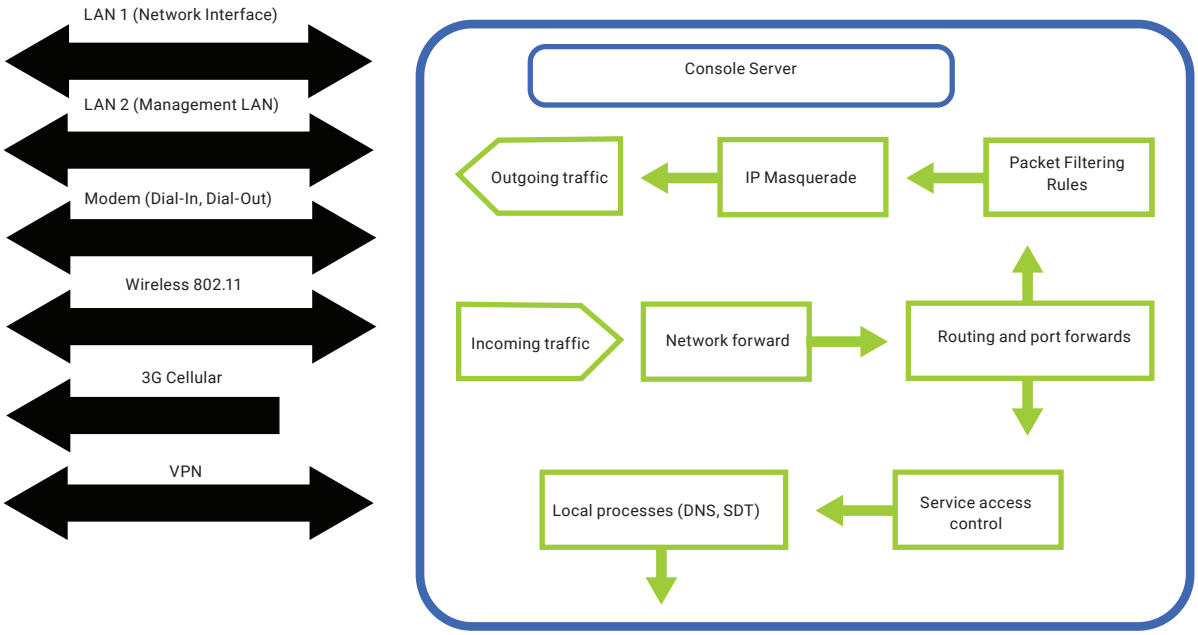


FIGURE 6-18.

When using IP Masquerading, devices on the external network cannot initiate connections to devices on the internal network. Port Forwards allows external users to connect to a specific port on the external interface of the console server and be redirected to a specified internal address for a device on the internal network.

With Firewall Rules, packet filtering inspects each packet passing through the firewall and accepts or rejects it based on user-defined rules.

Then Service Access Rules can be set for connecting to the console server/router itself.

6.8.1 CONFIGURING NETWORK FORWARDING AND IP MASQUERADING

To use a console server as an Internet or external network gateway requires establishing an external network connection and then setting up forwarding and masquerading.

By default, all console server models are configured so that they will not route traffic between networks. To use the console server as an Internet or external network gateway, forwarding must be enabled so that traffic can be routed from the internal network to the Internet or an external network.

NOTE: Network forwarding allows the network packets on one network interface (for example, LAN1/eth0) to be forwarded to another network interface (for example LAN2/eth1 or dial-out/cellular). Locally networked devices can IP-connect through the console server to devices on a remote network. IP masquerading is used to allow all the devices on your local private network to hide behind and share the one public IP address when connecting to a public network. This type of translation is only used for connections originating within the private network destined for the outside public network, and each outbound connection is maintained by using a different source IP port number.

- Navigate to System > Firewall.

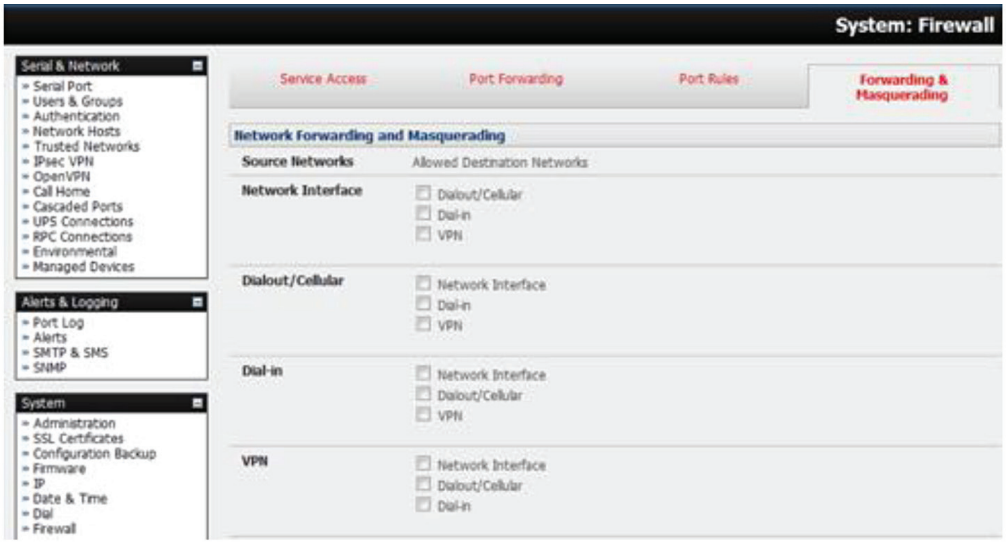


FIGURE 6-19.

Select the Forwarding & Masquerading tab.

- Find the Source Network to be routed and tick the relevant Destination Network.

  For example, to configure a single-Ethernet device such as the LES1204A-G as a cellular router, set:

  - the Source Network to Network Interface

  - the Destination Network to Dialout/Cellular.

IP Masquerading is generally required if the console server will be routing to the Internet, or if the external network being routed to does not have routing information about the internal network behind the console server.

IP Masquerading performs Source Network Address Translation (SNAT) on outgoing packets to make them appear like they've come from the console server rather than devices on the internal network.

When response packets come back devices on the external network, the console server translates the packet address back to the internal IP, so that it is routed correctly. This allows the console server to provide full outgoing connectivity for internal devices using a single IP Address on the external network.

By default, IP Masquerading is disabled for all networks. To enable masquerading:

- Navigate to System > Firewall.
- Select the Forwarding & Masquerading tab.
- Check Enable IP Masquerading (SNAT) on the network interface where masquerading is to be enabled.

6.8.2 CONFIGURING CLIENT DEVICES

Client devices on the local network must be configured with Gateway and DNS settings. This can be done statically on each device, or by using DHCP.

Manual configuration

Manually set a static gateway address (the address of the console server) and set the DNS server address to be the same as used on the external network. That is, if the console server is acting as an internet gateway or a cellular router, use the ISP-provided DNS server address.

DHCP configuration

- Navigate to System > IP.
- Click the tab of the interface connected to the internal network.

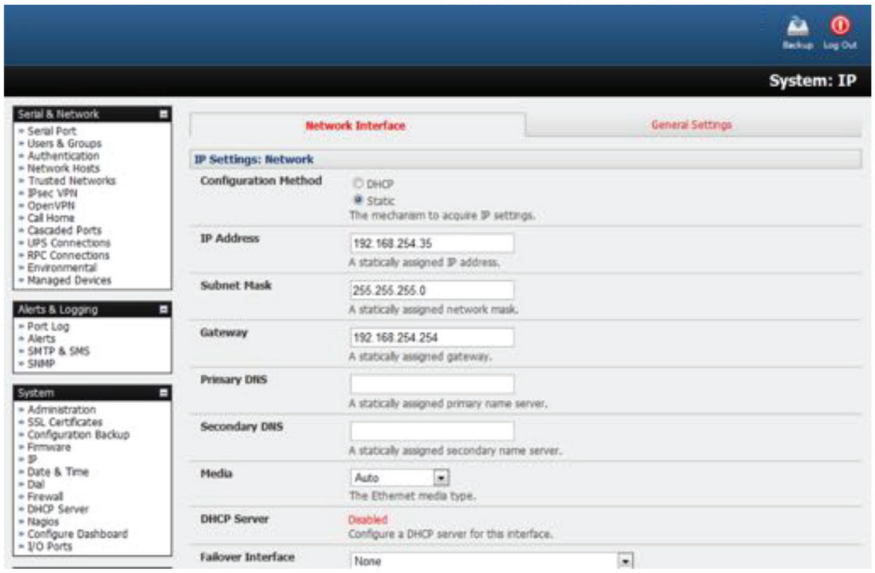


FIGURE 6-20.

- To use DHCP, a static address must be set: check that the IP Address and Subnet Mask fields have specific and static values entered.
- Click the Disabled link adjacent the DHCP Server entry.

  System > DHCP Server will load.
- Check the DHCP Server checkbox.
- Check the Use interface address as gateway checkbox.
- Set the Primary DNS and Secondary DNS addresses to the same addresses as are used on the external network.

If the console server is acting as an internal gateway or a cellular router, use the ISP-provided DNS server addresses.

- Enter the Default Lease time in seconds.
- Enter the Maximum Lease time in seconds.

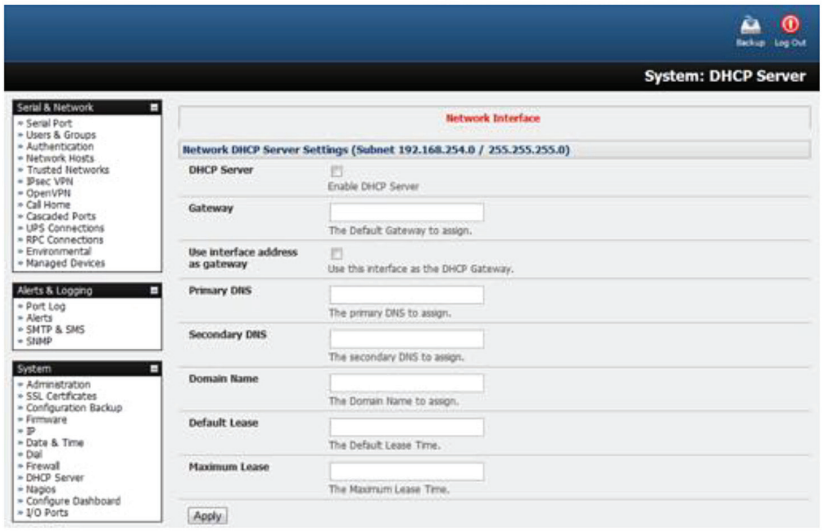


FIGURE 6-21.

Lease times are the number of seconds a dynamically assigned IP address is valid before the client must request it again.

- Click Apply.

  The DHCP server issue IP addresses sequentially from a specified address pool or pools.
- Click Add in the Dynamic Address Allocation Pool section.

FIGURE 6-22.

- Enter the DHCP Pool Start Address.
- Enter the DHCP Pool End Address.
- Click Apply.

6.8.3 PORT AND PROTOCOL FORWARDING

When using IP Masquerading, devices on the external network cannot initiate connections to devices on the internal network.

To work around this, Port Forwards can be set up to allow external users to connect to a specific port, or range of ports on the external interface of the console server or cellular router. Port forwarding also allows the console server or cellular router to redirect data to a specified internal address and port range.

To setup a port and protocol forward:

- Navigate to System > Firewall.

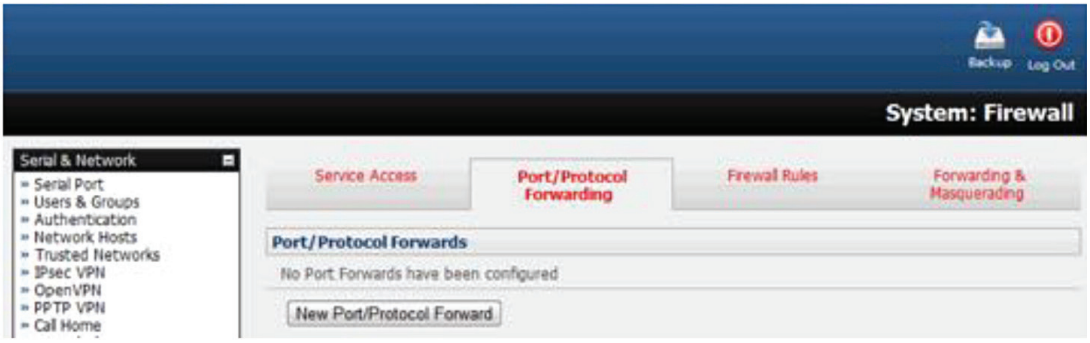


FIGURE 6-23.

- Click the Port/Protocol Forwarding tab.
- Click New Port/Protocol Forward.
- Fill in the following fields.

TABLE 6-7. PORT/PROTOCOL FORWARDING FIELDS

FIELD	PURPOSE
Name	Name for the port forward. This should describe the target and the service that the port forward is used to access.
Input Interface	This allows the user to only forward the port from a specific interface. In most cases, this should be left as Any.
Source Address/Address Range	This allows the user to restrict access to a port forward to a specific source IP address or IP address range of the data. This may be left blank. IP address ranges use the format ip/netmask (where netmask is in bits 1-32).
Destination Address/Address Range	The destination IP address/address range to match. This may be left blank. IP address ranges use the format ip/netmask (where netmask is in bits 1-32)
Input Port Range	The range of ports to forward to the destination IP. These will be the port(s) specified when accessing the port forward. These ports need not be the same as the output port range.
Protocol	The protocol of the data being forwarded. The options are TCP, UDP, TCP and UDP, ICMP, ESP, GRE, or Any.
Output Address	The target of the port forward. This is an address on the internal network where packets sent to the Input Interface on the input port range are sent.
Output Port Range	The port or range of ports that the packets will be redirected to on the Output Address. Ranges use the format start-finish. Only valid for TCP and UDP protocols.

For example, to forward port 8443 to an internal HTTPS server on 192.168.10.2, use the following settings:

TABLE 6-8. PORT/PROTOCOL FORWARDING EXAMPLE

FIELD	DESCRIPTION
Name	Administrator's choice
Input Interface	Any
Source Address/Address Range	Leave blank
Destination Address/Address Range	Leave blank
Input Port Range	8443
Protocol	TCP
Output Address	192.168.10.2
Output Port Range	443

6.8.4 FIREWALL RULES

Firewall rules can be used to block or allow traffic through an interface based on port number, the source IP address, the destination IP address or range, the direction (ingress or egress), the protocol or any combination of these. This can be used to allow custom on-box services, or block traffic based on policy.

To setup a firewall rule:

- Navigate to System > Firewall.
- Click the Firewall Rules tab.

NOTE: Prior to firmware v3.4, this tab was labeled Port Rules and fewer firewall rules could be configured.

- Click New Firewall Rule.
- Fill in the following fields.

TABLE 6-9. FIREWALL RULE FIELDS

FIELD	PURPOSE
Name	Name the rule. This name should describe the policy the firewall rule is being used to implement (for example, Block FTP or Allow Tony).
Interface	Select the interface that the firewall rule will be applied to. Choices include Any, Dialout/Cellular, VPN, Network Interface, and Dial-in.
Port Range	Specify the Port or range of Ports (for example 1000 – 1500) that the rule will apply to. This may be left blank for Any.
Source MAC Address	Specify the source MAC address to be matched. This may be left blank for Any. MAC addresses use the format XX:XX:XX:XX:XX:XX, where XX are hex digits
Source Address Range	Specify the source IP address (or address range) to match. IP address ranges use the format ip/netmask (where netmask is in bits 1-32). This may be left blank for Any..

TABLE 6-9 (CONTINUED). FIREWALL RULE FIELDS

FIELD	PURPOSE
Destination Range	Specify the destination IP address/address range to match. IP address ranges use the format ip/netmask (where netmask is in bits 1-32). This may be left blank.
Protocol	Select if the firewall rule will apply to TCP, UDP, TCP and UDP, ICMP, ESP, GRE, or Any.
Direction	Select the traffic direction that the firewall rule will apply to: Ingress = incoming; Egress = outgoing.
Action	Select the action (Accept or Block) to be applied to the packets detected that match the Interface + Port Range + Source Address + Destination Range+ Protocol+ Direction.

For example, to block all SSH traffic from leaving a Dialout Interface, use the following settings.

TABLE 6-10. FIREWALL RULE EXAMPLE #1

FIELD	PURPOSE
Name	Administrator's choice
Interface	Dialout/Cellular
Port Range	22
Source MAC Address	Left blank
Source Address Range	Left blank (Any)
Destination Range	Left blank
Protocol	TCP
Direction	Egress
Action	Block

Firewall rules are processed in a set order, from top to bottom. So rule placement is important.

For example, with the following rules, all traffic coming in over the Network Interface is blocked except when it comes from two nominated IP addresses (SysAdmin and Tony):

TABLE 6-11. FIREWALL RULE EXAMPLE #2

FIELD	VALUE	VALUE	VALUE
Interface	Any	Any	Network Interface
Port Range	Any	Any	Any
Source MAC Address	Any	Any	Any
Source Address Range	SysAdmin's IP address	Tony's IP address	Any
Destination Range	Any	Any	Any
Protocol	TCP	TCP	TCP
Direction	Ingress	Ingress	Ingress
Action	Accept	Accept	Block

If the Rule Order is changed so the Block Everyone Else rule was second on the list, Tony's traffic—coming in over the Network Interface—would be blocked.

6.8.5 PACKET STATE MATCHING IN FIREWALL RULES

As of firmware 4.0.0, Firewall rules can include packet state matching.

This is implemented using an iptables extension module and can be set as follows:

- Navigate to System > Firewall > Firewall Rules.
- In either the IPv4 or IPv6 section, click the New Firewall Rule button.
- Enter a Name for the new rule in the Name field.
- Select the Interface the new rule will be applied against from the Interface pop-up menu.

NOTE: The available interfaces vary depending on the exact hardware available on the console server but, by default, new firewall rules are applied against Any (i.e., all) available interface.

- If the selected interface operates the TCP or UDP protocol, enter a port or port range of the rule's destination.
- If the firewall rule is to apply against a particular MAC address, enter this value in the Source MAC address field.

  MAC addresses must be entered in standard xx:xx:xx:xx:xx:xx format (where each xx is a hexadecimal value).
- If the firewall rule is to apply against a particular source address or range of source addresses, enter this address or address range in the Source Address/Address Range field.

  Address ranges can be entered using the ip-address/netmask syntax.
- If the firewall rule is to apply to a particular destination address or address range, enter this address or address range in the Destination Address/Address Range field.

  As with the Source Address/Address Range field, address ranges can be entered using the ip-address/netmask syntax.
- Set the data protocol against which the firewall rule will apply.

  By default, new firewall rules apply against the TCP protocol.
- Set the direction of data travel against which the firewall rule will apply.

  This setting can take one of two values: Ingress or Egress. The default is Ingress.

  Ingress means data arriving at an interface from elsewhere. Egress means data leaving an interface and going to elsewhere.
- Select the desired packet state to match against from the Connection State pop-up menu.

  Available options are New, Established/Related, and Any.

  The default option is Any.

NOTE: The default option leaves packet state matching inactive. With this option, no extra specifications are added to the firewall rule.

- Select the desired action to be taken regarding packets of the chosen state from the Action pop-up menu.

  The two available options are Block and Accept.

  The default action is Block.
- Click the Save button.

Using the Connection State pop-up menu in System > Firewall > Firewall Rules > IPv4 > New Firewall Rule to set packet state matching to New or Established/Related is equivalent to running one of the following at a shell-prompt:

```
# iptables -m state --state NEW
# iptables -m state --state ESTABLISHED,RELATED
```

For example:

```
# iptables -I INPUT -p tcp --dport 23 -m state --state \ ESTABLISHED,RELATED -j ACCEPT
```

This tells the firewall to accept incoming Telnet traffic for previously established Telnet sessions.

If the rule is created in IPv6 > New Firewall Rule, it is the equivalent of running one of the following at a shell-prompt:

```
# ip6tables -m state --state NEW
# ip6tables -m state --state ESTABLISHED,RELATED
```

For example:

```
# ip6tables -I INPUT -p tcp --dport 23 -m state --state \ ESTABLISHED,RELATED -j ACCEPT
```

As with the iptables example, this tells the firewall to accept incoming Telnet traffic for previously established Telnet sessions.

For more on iptables, ip6tables and iptables-extensions, see the respective manual pages: iptables, ip6tables and iptables-extensions.

CHAPTER 7: SSH TUNNELS AND SDT CONNECTOR

Each Black Box console server has an embedded SSH server and uses SSH tunneling so remote users can securely connect through the console server to Managed Devices using text-based console tools (such as SSH, telnet, SoL) or graphical tools (such VNC, RDP, HTTPS, HTTP, X11, VMware, DRAC, iLO).

The Managed Devices being accessed can be located on the same local network as the console server or they can be attached to the console server via a serial port. The remote User/Administrator connects to the console server thru an SSH tunnel via dial-up, wireless or ISDN modem; a broadband Internet connection; the enterprise VPN network or the local network.

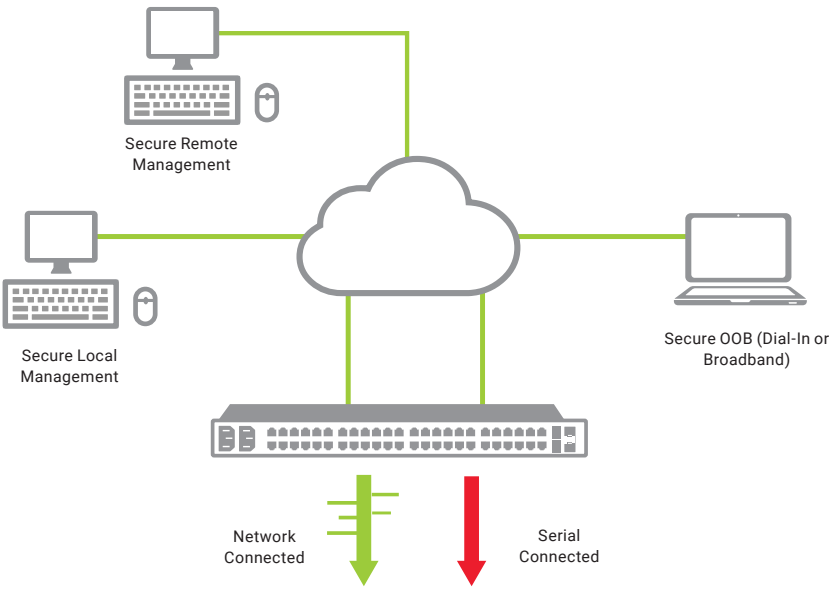


FIGURE 7-1.

To set up the secure SSH tunnel from the Client PC to the console server, you must install (if necessary) and launch SSH client software on the User's or Administrator's PC.

Black Box recommends that you use the SDT Connector client software that is supplied with the console server for this. SDT Connector is simple to install and auto-configure and it will provide all your users with point-and-click access to all the systems and devices in the secure network.

With one click, SDT Connector sets up a secure SSH tunnel from the client to the selected console server, then establishes a port forward connection to the target network connected host or serial connected device, then executes the client application that will be used in communicating with the host.

This chapter details the basic SDT Connector operations:

- Configuring the console server for SSH tunneled access to network attached hosts and setting up permitted Services and user access.
- Setting up the SDT Connector client with gateway, host, service and client application details and making connections between the Client PC and hosts connected to the console server.
- Using SDT Connector to browser access the Management Console.
- Using SDT Connector to Telnet or SSH connect to devices that are serially attached to the console server.

CHAPTER 7: SSH TUNNELS AND SDT CONNECTOR

The chapter then covers more advanced SDT Connector and SSH tunneling topics:

- Using SDT Connector for out-of-band access.
- Automatic importing and exporting of configurations.
- Configuring Public Key Authentication.
- Setting up an SDT Secure Tunnel for Remote Desktop.
- Setting up an SDT Secure Tunnel for VNC.
- Using SDT to IP connect to hosts that are serially attached to the console server.

7.1 CONFIGURING FOR SSH TUNNELING TO HOSTS

To set up the console server for SSH tunneled access a network attached host:

- Add the new host and the permitted services using the Serial & Network > Network Hosts menu as detailed in Network Hosts (Section 5.4). Only these permitted services will be forwarded through by SSH to the host. All other services (TCP/UDP ports) will be blocked.

  Following are some of the TCP Ports used by SDT in the console server.

TABLE 7-1. TCP PORTS USED BY SDT

PORT	APPLICATION	NOTES
22	SSH	All SDT tunneled connections
23	Telnet	On local LAN. Forwarded inside tunnel
80	HTTP	On local LAN. Forwarded inside tunnel
3389	RDP	On local LAN. Forwarded inside tunnel
5900	VNC	On local LAN. Forwarded inside tunnel
73xx	RDP over serial	From local LAN. xx is the serial port # (eg 7301–7348 on a 48-port console server
79xx	VNC over serial	From local LAN. xx is the serial port # (eg 7301–7348 on a 48-port console server

- Add the new Users using Serial & Network > Users & Groups menu as detailed in Network Hosts (Section 5.4). Users can be authorized to access the console server ports and specified network-attached hosts.

FIGURE 7-2. NETWORK HOSTS SCREEN

To simplify configuration, the Administrator can first set up Groups with group access permissions, then Users can be classified as members of particular Groups.

7.2 SDT CONNECTOR CLIENT CONFIGURATION

The SDT Connector client works with all Black Box console servers. Each remote console server has an embedded OpenSSH based server which can be configured to port forward connections from the SDT Connector client to hosts on their local network (see Chapter 6).

The SDT Connector can also be pre-configured with the access tools and applications that will be available to be run when access to a particular host has been established.

SDT Connector can connect to the console server using an alternate OOB access. It can also access the console server itself and devices connected to the console server's serial ports.

7.2.1 SDT CONNECTOR CLIENT INSTALLATION

SDT Connector's set up tool, SDTConnector Setup-1.n.exe or sdtcon-1.n.tar.gz, is on the CD supplied with your console server. It can also be downloaded from blackbox.com

To install, run the set-up program.

On Windows, SDTConnectorSetup-1.n.exe installs SDT Connector 1.n.exe and the config file defaults.xml. If defaults.xml exists, it is not overwritten. To remove earlier config files, run regedit, search for SDT Connector and remove the directory with this name.

For Linux and other Unix clients, SDTConnector.tar.gz will install the sdtcon-1.n.jar and the config file defaults.xml.

FIGURE 7-3. SDT CONNECTOR SETUP WIZARD SCREEN

Once complete you will have SDT Connector on your machine and an icon on your desktop. To launch the SDT Connector client, double-click this icon.

FIGURE 7-4. SDT CONNECTOR ICON

NOTE: SDT Connector is a Java application. It must have a Java Runtime Environment (JRE) installed. It will install on Windows 2000 and later and on most Linux platforms. Solaris platforms are also supported however they must have Firefox installed. SDT Connector can run on any system with Java 1.4.2 and above installed, but it assumes the web browser is Firefox, and that xterm -e telnet opens a telnet window.

To operate SDT Connector, you first add new gateways to the client software by entering the access details for each console server (see Section 7.2.2), then let the client auto-configure with all host and serial port connections from each console server (see Section 7.2.3), then point-and-click to connect to the Hosts and serial devices (see Section 7.2.4).

Alternately you can manually add network connected hosts (see Section 7.2.5) and manually configure new services to be used in accessing the console server and the hosts (see Section 7.2.6), then manually configuring clients to run on the PC that will use the service to connect to the hosts and serial port devices (see Section 7.2.7). SDT Connector can also be set up to make an out-of-band connection to the console server.

7.2.2 CONFIGURING A NEW GATEWAY IN THE SDT CONNECTOR CLIENT

To create a secure SSH tunnel to a new console server:

- Select File > New Gateway or click the New Gateway icon.

FIGURE 7-5.

- Enter the IP address or hostname of the console server.
- Enter the SSH port (typically port 22).

If SDT Connector is connecting to a remote console server through the public Internet or a routed network, you will need to:

- Determine the public IP address of the console server or the public IP address of the router or firewall that connects the console server to the Internet.

One way to find the public IP address is to access / or / from a computer on the same network as the console server and note the reported IP address.

- Set up port-forwarding for TCP port 22 on any firewall, router or NAT service located between SDT Connector and the console server.

  http://www.portforward.com/ has port-forwarding instructions for a range of routers. The Open Port Check tool from http://www.canyouseeme.org/ can be used to check if port-forwarding through a firewall, router or NAT service has been properly configured.
- Enter the Username and Password of a user on the gateway who has been enabled to connect via SSH.
- Eptionally, enter a Descriptive Name to display instead of the IP address or hostname.
- Eptionally enter desired information in the Description/Notes field.

  For example: the console server's site location; the console server's running firmware version; or details on the site's network configuration.

- Click OK.

  The new gateway will appear in the SDT Connector home page.

NOTE: For an SDT Connector user to access a console server and then access specific hosts or serial devices connected to that console server, that user must first be set up on the console server, and must be authorized to access the specific ports on the specific hosts (see Chapter 6). Only these permitted services will be forwarded through by SSH to the Host. All other services (TCP/UDP ports) are blocked.

7.2.3 AUTO-CONFIGURE SDT CONNECTOR CLIENT WITH THE USER'S ACCESS PRIVILEGES

Each user on the console server has an access profile which has been configured with those specific connected hosts and serial port devices the user has authority to access, and a specific set of the enabled services for each of these. This configuration can be auto-uploaded into the SDT Connector client.

- Select File > New Gateway (or click the New Gateway icon).
- Click Retrieve Hosts.

SDT Connector will:

- Configure access to network connected Hosts that the user is authorized to access and will, for each of these Hosts, set up the services (for example, HTTPS, IPMI2.0) and the related IP ports being redirected.

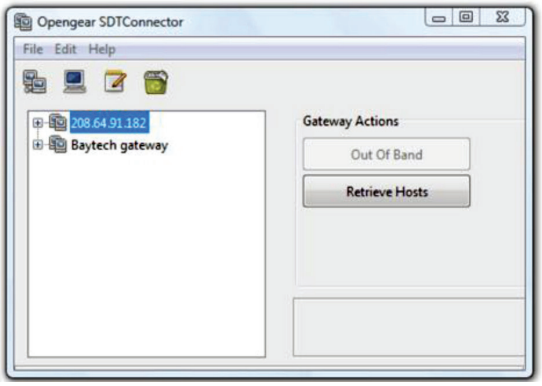


FIGURE 7-6. RETRIEVE HOSTS SCREEN

- Configure access to the console server itself. This is shown as a Local Services host.
- Configure access with the enabled services for the serial port devices connected to the console server.

FIGURE 7-7. SERVICES SCREEN

NOTE: Retrieve Hosts auto-configures all classes of user whether they are members of user, admin, some other group, or no group. SDT Connector will not auto-configure the root. We recommend that root only be used for initial config and for adding an initial admin account to the console server.

7.2.4 MAKE AN SDT CONNECTION THROUGH THE GATEWAY TO A HOST

- Select the host to be accessed.
- Click the Service to be used in accessing that host.

  The SSH tunnel to the gateway is established, the appropriate ports redirected through to the host, and the appropriate local client application is launched pointing at the local endpoint of the redirection.

The SDT Connector client can be configured with an unlimited number of gateways and each gateway can be configured to port forward to an unlimited number of locally networked Hosts. Similarly there is no limit on the number of SDT Connector clients who can be configured to access the one Gateway. Nor are there limits on the number of Host connections that an SDT Connector client can concurrently have open through the one Gateway tunnel.

FIGURE 7-8.

There is a limit to the number of SDT Connector SSH tunnels that can be open at the one time on a particular Gateway. LES1400, LES1200 and LES1508A models each support at least 50 such concurrent connections. For a site with a LES1400 gateway you can have, at any time up to 50 users securely controlling an unlimited number of network attached computers and appliances (servers, routers, etc.) at that site. LES1600, LES1700-R2 and LES1516A, LES1532A, LES1548A support many hundreds of simultaneous client tunnels.

7.2.5 MANUALLY ADDING A HOST TO THE SDT CONNECTOR GATEWAY

For each gateway, you can manually specify the network connected hosts that will be accessed through that console server and for each host, specify the services that will used in communicating with the host.

- Select File > New Host (or select a gateway and click the Host icon).

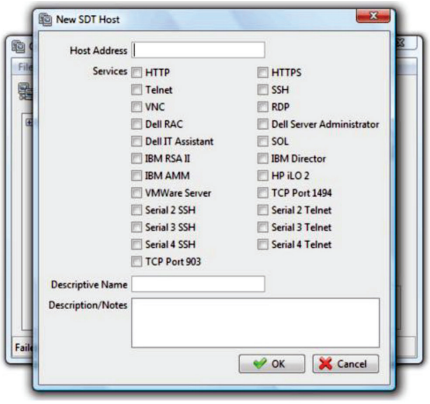


FIGURE 7-9. NEW SDT HOST SCREEN

- Enter the IP address or hostname of the host.

NOTE: A hostname must be resolvable by the gateway.

- Select which Services are to be used in accessing the new host.

  A range of service options are pre-configured in the default SDT Connector client (RDP, VNC, HTTP, HTTPS, Dell RAC, VMware etc). If you wish to add services beyond the pre-configured range, see the next section.

- Optionally, enter a Descriptive Name to display instead of the IP address or hostname.
- Optionally enter desired information in the Description/Notes field.

  For example: the console server's site location; the console server's running firmware version; or details on the site's network configuration.

- Click OK.

7.2.6 MANUALLY ADDING NEW SERVICES TO THE NEW HOSTS

To extend the range of services that can be used when accessing hosts with SDT Connector:

- Select Edit > Preferences.

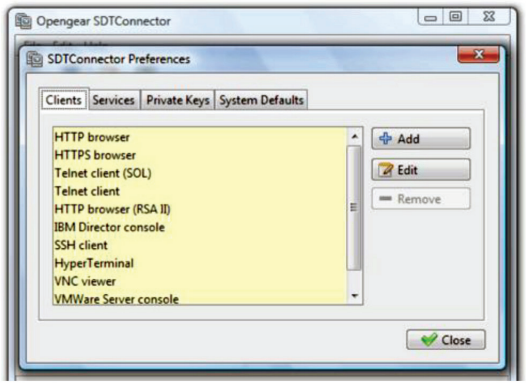


FIGURE 7-10. EDIT PREFERENCES SCREEN

- Click the Services tab.
- Click Add.
- Enter a Service Name.
- Click Add.
- Under the General tab, enter the TCP Port that this service runs on (for example, port 80 for HTTP).
- Optionally, select the client to use to access the local endpoint of the redirection.
- Select which client application is associated with the new service.

  A range of client application options are pre-configured in the default SDT Connector (RDP client, VNC client, HTTP browser, HTTPS browser, Telnet client etc). If you wish to add new client applications to this range proceed to the next section.

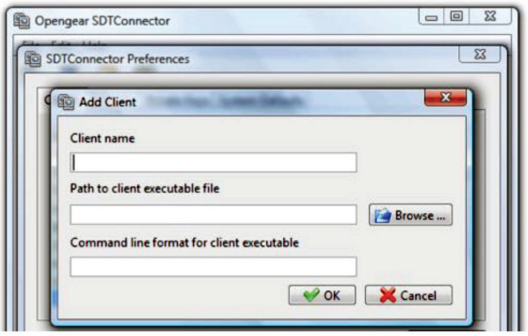


FIGURE 7-11. ENTER CLIENT NAME SCREEN

- Click OK.
- Click Close.

A service typically consists of a single SSH port redirection and a local client to access it. It may consist of several redirections, some or all of which may have clients associated with them.

An example is the Dell RAC service. The first redirection is for the HTTPS connection to the RAC server. It has a client associated with it (web browser) that is launched immediately upon clicking the button for this service.

The second redirection is for the VNC service that the user may choose to later launch from the RAC web console. It automatically loads in a Java client served through the web browser, so it does not need a local client associated with it.

On the Add Service screen, you can click Add as many times as needed to add multiple new port redirections and associated clients.

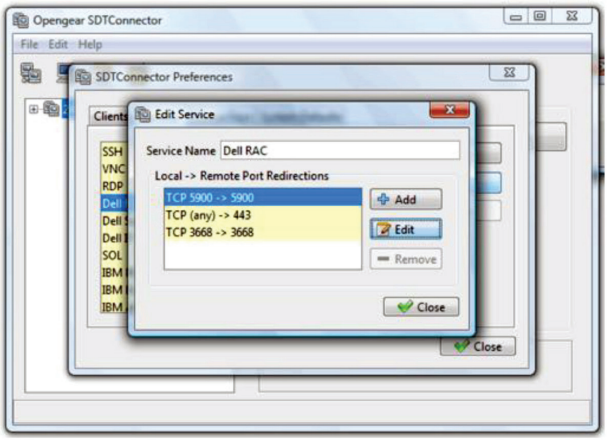


FIGURE 7-12. EDIT SERVICE SCREEN

You may also specify Advanced port redirection options:

- Enter the local address to bind to when creating the local endpoint of the redirection.

  It is not usually necessary to change this from localhost.
- Enter a local TCP port to bind to when creating the local endpoint of the redirection.

  If this is left blank, a random port will be selected.

NOTE: SDT Connector can also tunnel UDP services. SDT Connector tunnels the UDP traffic through the TCP SSH redirection, so in effect it is a tunnel within a tunnel. Enter the UDP port on which the service is running on the host. This will also be the local UDP port that SDT Connector binds as the local endpoint of the tunnel. For UDP services, you still need to specify a TCP port under General. This will be an arbitrary TCP port that is not in use on the gateway. An example of this is the SOL Proxy service. It redirects local UDP port 623 to remote UDP port 623 over the arbitrary TCP port 6667.

7.2.7 ADDING A CLIENT PROGRAM TO BE STARTED FOR THE NEW SERVICE

Clients are local applications that may be launched when a related service is clicked. To add to the pool of client programs:

- Select Edit > Preferences.

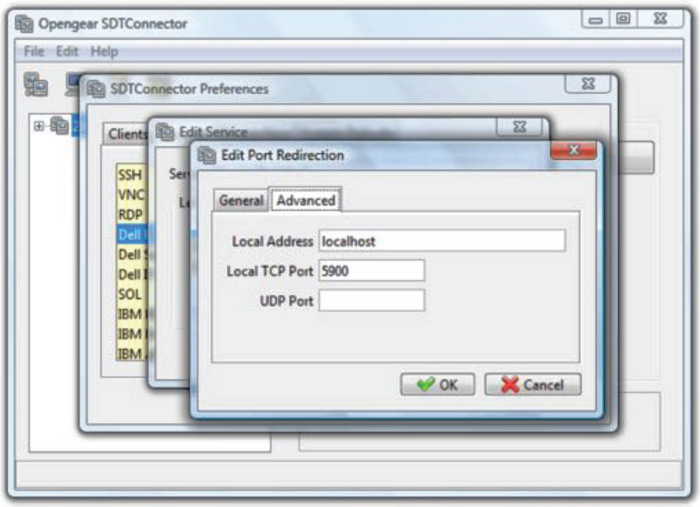


FIGURE 7-13. EDIT PORT REDIRECTION, ADVANCED TAB

- Click the Client tab.
- Click Add.
- Enter a Client name.
- Enter the Path to the client executable file or click Browse to locate the client application.
- Enter a Command line format for client executable associated with launching the client.

  SDT Connector typically launches a client using command line arguments to point it at the local endpoint of the redirection. Three keywords specify the command line format. When launching the client, SDT Connector substitutes these keywords with appropriate values.

TABLE 7-2. KEYWORDS

KEYWORD	DESCRIPTION
%path%	The path to the executable file. Takes the previous field value: Path to the client executable file.
%host%	The local address to which the local endpoint of the redirection is bound. That is, the Local Address field for the Service redirection Advanced options.
%port%	The local port to which the local endpoint of the redirection is bound. That is, the Local TCP Port field for the Service redirection Advanced options. If port is unspecified (Any) an appropriate randomly selected port is substituted.

For example, SDT Connector is preconfigured for Windows with an HTTP client that connects to the Windows user's default browser. If there is no default browser, Firefox is used.

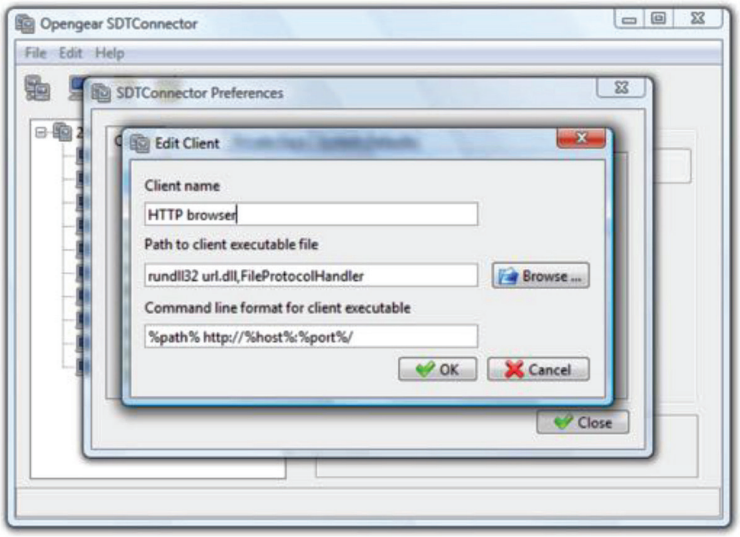


FIGURE 7-14. EDIT CLIENT SCREEN

Some clients are launched in a command line or terminal window. The Telnet client, for example. In this case, Path to client executable file is telnet and the Command line format for client executable is cmd /c start %path% %host% %port%.

- Click OK.

7.2.8 DIAL-IN CONFIGURATION

If the client is dialing into the console server's Local/Console port, set up a dial-in PPP link.

- Configure the console server for dial-in access, following the steps in Section 6.1.
- Set up the PPP client software on the remote computer, following the steps in Chapter 6.

Once you have a dial-in PPP connection established, set up the secure SSH tunnel from the remote computer to the console server.

7.3 SDT CONNECTOR TO MANAGEMENT CONSOLE

SDT Connector can also be configured for browser access to the gateway's Management Console and for Telnet or SSH access to the gateway's shell. For these connections to the gateway itself, you must configure SDT Connector to access the gateway by setting the console server up as a host, and then configuring the appropriate services.

- Launch SDT Connector on your PC.
- Assuming you have already set up the console server as a Gateway in your SDT Connector client (with username, password, etc.), select this newly added Gateway and click the Host icon to create a host.
- Alternatively, select File > New Host.
- Enter 127.0.0.1 as the Host Address.
- Optionally add details in the Descriptive Name and Description/Notes fields.
- Click OK.
- Click the HTTP or HTTPS services icon to access the gateway's Management Console.

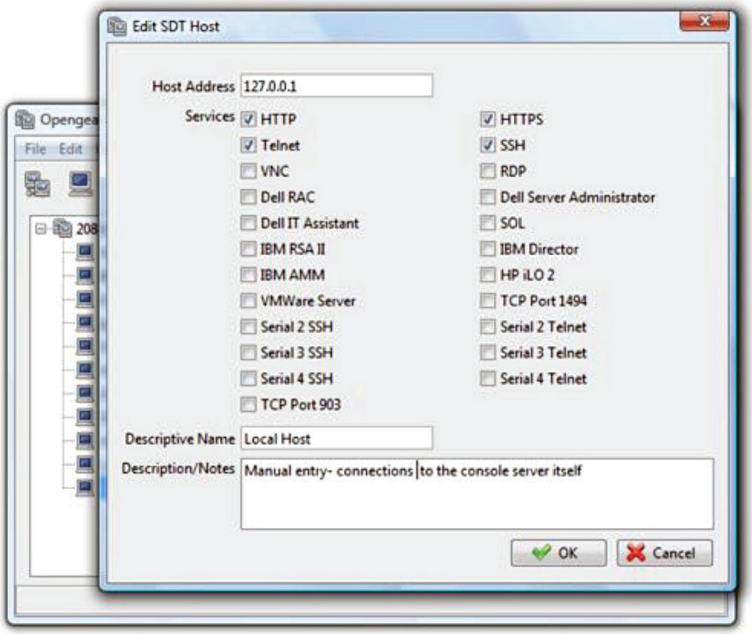


FIGURE 7-15. EDIT SDT HOST SCREEN

Click SSH or Telnet to access the gateway's command line console.

To enable SDT access to the gateway console, you must configure the console server to allow port forwarded network access to itself.

As of firmware v3.3, this can be done using the console server's Management Console.

- Navigate to System > Firewall.
- Click the Service Access tab.
- Enable SSH Command Shell access on the Network Interface and on any Out-of-band Interfaces.

With firmware versions prior to v3.3, do the following.

- Navigate to Serial & Network > Network Hosts.
- Click Add Host.
- In the IP Address/DNS Name field enter 127.0.0.1.

  This is the loopback address.
- Enter Loopback in the Description field.
- Remove all entries under Permitted Services except for those that will be used in accessing the Management Console (80/http or 443/https) or the command line (22/ssh or 23/telnet).
- Click Apply.

By default, Administrators have gateway access privileges. For Users to access the gateway Management Console, the required access privileges must be granted.

- Navigate to Serial & Network > Users & Groups.
- Click Add User.
- Enter a Username, Description and Password.
- Select 127.0.0.1 from the Accessible Host(s) pop-up menu.
- Click Apply.

7.4 SDT CONNECTOR: TELNET OR SSH CONNECT TO SERIALLY-ATTACHED DEVICES

SDT Connector can also be used to access text consoles on devices that are attached to the console server's serial ports. For these connections, configure the SDT Connector client software with a Service that will access the target gateway serial port, and then set the gateway up as a host.

- Launch SDT Connector on your PC.
- Select Edit > Preferences.
- Click Add.
- Enter Serial Port 2 as the Service Name.
- Click Add.
- Select Telnet as the Client.
- Enter 2002 as the TCP Port.
- Click OK.
- Close the Add Service window.
- Close the SDTConnector Preferences window.
- Assuming you have already set up the console server as a Gateway in your SDT Connector client (with username, password, etc.), select this newly added Gateway and click the Host icon to create a host.
- Alternatively, select File > New Host.
- Enter 127.0.0.1 as the Host Address.
- Optionally, add details in the Descriptive Name and Description/Notes fields.
- Click OK.
- Click the Serial Port 2 icon for Telnet access to the serial console on the device attached to serial port #2 on the gateway.

To enable SDT Connector to access to devices connected to the gateway's serial ports, configure the console server to allow port forwarded network access to itself, and enable access to the nominated serial port.

- Navigate to Serial & Network > Serial Port.
- Click Edit next to the selected Port.

  For example, click Edit next to Port 2 if the target device is attached to the second serial port.
- Ensure the port's serial configuration is appropriate for the attached device.
- Set the Console Server Setting to Console Server Mode.
- Click Apply.
- Navigate to Serial & Network > Network Hosts.
- Click Add Host.
- In the IP Address/DNS Name field enter 127.0.0.1.

  This is the loopback address.
- Enter Loopback in the Description field.
- Remove all entries under Permitted Services.
- Select TCP.
- Enter 200n in the Port field.

NOTE: "n" corresponds to the Serial Port selected in the step above. For Serial Port 2, for example, enter 2002.

- Click Add.
- Click Apply.

By default, Administrators have gateway and serial port access privileges. For Users to access the gateway Management Console and the serial port, the required access privileges must be granted.

- Navigate to Serial & Network > Users & Groups.
- Click Add User.
- Enter a Username, Description and Password.
- Select 127.0.0.1 from the Accessible Host(s) pop-up menu.
- Select Port 2 from the Accessible Port(s) pop-up menu.
- Click Apply.

7.5 USING SDT CONNECTOR FOR OUT-OF-BAND CONNECTION TO THE GATEWAY

SDT Connector can also be set up to connect to the console server out-of-band (OOB). OOB access uses an alternate path for connecting to the console server to that used for regular data traffic. OOB access is useful for when the primary link into the console server is unavailable or unreliable.

Typically, a console server's primary link is a broadband Internet connection or Internet connection via a LAN or VPN, and the secondary out-of-band connectivity is provided by a dial-up or wireless modem directly attached to the console server.

Out-of-band access enables you to access the hosts and serial devices on the network, diagnose any connectivity issues, and restore the console server's primary link.

In SDT Connector, OOB access is configured by providing the secondary IP address of the gateway, and telling SDT Connector how to start and stop the OOB connection. Starting an OOB connection may be achieved by initiating a dial up connection, or adding an alternate route to the console server. SDT Connector allows for maximum flexibility in this regard, by allowing you to provide your own scripts or commands for starting and stopping the OOB connection.

To configure SDT Connector for OOB access:

- Choose File > New Gateway.
- Click the Out Of Band tab.
- Enter the console server's Out-of-Band IP address in the Secondary Address field.

The console server's Out-of-Band IP address is the address the console server is accessible from when using the Out-of-Band access route.

- Change the Port value if the console server is using a port other than the default 22 for SSH access.
- Enter the command or path to a script to start the OOB connection in the Start Command field.
- To initiate a pre-configured dial-up connection under Windows, use the following Start Command string:

  cmd /c start "Starting Out of Band Connection" /wait /min rasdial network_connection login password

  where network_connection is the name of the network connection as displayed in Control Panel > Network Connections, login is the console server's dial-in username, and password is the console server's dial-in password.
- To initiate a pre-configured dial-up connection under Linux, use the following Start Command string:

  pon network_connection

  where network_connection is the name of the connection.
- Enter the command or path to a script to stop the OOB connection in the Stop Command field.

- To stop a pre-configured dial-up connection under Windows, use the following Stop Command string:

  cmd /c start "Stopping Out of Band Connection" /wait /min rasdial network_connection /disconnect

  where network_connection is the name of the network connection as displayed in Control Panel > Network Connections.
- To stop a pre-configured dial-up connection under Linux, use the following Stop Command string:

  poff network_connection

To make the OOB connection using SDT Connector:

- Select the console server to connect to.
- Click the Out Of Band button.

  The status bar changes color to indicate this console server is being accessed using the OOB link rather than the primary link.

When you connect to a service on a host behind the console server, or to the console server itself, SDT Connector will initiate the OOB connection using the provided Start Command. The OOB connection isn't stopped (using the provided Stop Command) until Out Of Band under Gateway Actions is clicked off, at which point the status bar will return to its normal color.

7.6 IMPORTING AND EXPORTING PREFERENCES

To enable the distribution of pre-configured client config files, SDT Connector has an Import and Export facility:

To save a configuration .xml file for backup or for importing into other SDT Connector clients):

- Select File > Export Preferences.
- Select the location to save the configuration file.

To import a configuration:

- Select File > Import Preferences.
- Select the .xml configuration file to be installed.

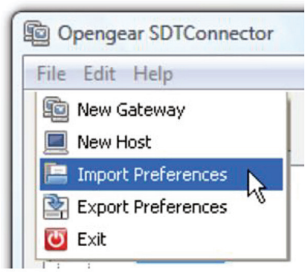


FIGURE 7-16. IMPORT PREFERENCES SCREEN

7.7 SDT CONNECTOR PUBLIC KEY AUTHENTICATION

SDT Connector can authenticate against an SSH gateway using your SSH key pair rather than requiring you to enter your password. This is known as public key authentication.

To use public key authentication with SDT Connector, first you must add the public part of your SSH key pair to your SSH gateway.

- Ensure the SSH gateway allows public key authentication.

  This is typically the default behavior.
- If you do not already have a public/private key pair for your client PC (the one running SDT Connector) generate them now using ssh-keygen, PuTTYgen or a similar tool.

See Section 16.6 for details on generating and installing public/private key pairs.

NOTE: You can use RSA or DSA. In this case, leave the passphrase field blank.

- Upload the public part of your SSH key pair (typically named id_rsa.pub or id_dsa.pub) to the SSH gateway, or otherwise add to .ssh/ authorized keys in your home directory on the SSH gateway.
- Add the private SSH key (typically named id_rsa or id_dsa) to SDT Connector.
- Click Edit > Preferences.
- Select Private Keys.
- Click Add.
- Navigate to and select the private key file.
- Click OK.

You do not have to add the public SSH key: it is calculated using the private key.

SDT Connector will now use public key authentication when connecting through the SSH console server.

NOTE: You may have to restart SDT Connector to shut down existing SSH tunnels established using password authentication.

If you have a host behind the console server that you connect to by clicking the SSH button in SDT Connector, you may wish to configure access to it for public key authentication as well.

This configuration is entirely independent of SDT Connector and the SSH console server. You must configure the SSH client that SDT Connector launches (for example Putty or OpenSSH) and the host's SSH server for public key authentication. Essentially, what you are using is SSH over SSH, and the two SSH connections are entirely separate.

7.8 SETTING UP SDT FOR REMOTE DESKTOP ACCESS

Microsoft's Remote Desktop Protocol (RDP) enables the system manager to:

- Securely access and manages remote Windows computers
- To reconfigure applications and user profiles on Windows computers
- To upgrade a Windows server operating system.
- Reboot the machine and more.

Secure Tunneling uses SSH tunneling, so this RDP traffic is securely transferred through an authenticated and encrypted tunnel.

SDT with RDP also allows remote Users to connect to Windows XP and later computers and to Windows 2000 Terminal Servers; and to have access to all of the applications, files, and network resources (with full graphical interface just as though they were in front of the computer screen at work).

To set up a secure Remote Desktop connection you must enable Remote Desktop on the target Windows computer that is to be accessed and configure the RPD client software on the client PC.

7.8.1 ENABLE REMOTE DESKTOP ON THE TARGET WINDOWS COMPUTER TO BE ACCESSED

NOTE: Windows XP Professional and Windows Vista only support one Remote Desktop session and it connects directly to the Windows root console. Windows Server 2003 supports three sessions: the console session and two other general sessions. Windows Server 2008 supports multiple sessions.

To enable Remote Desktop on the Windows computer being accessed:

- Navigate to Start Menu > Control Panel.
- Double-click the System icon.

- Click the Remote tab.

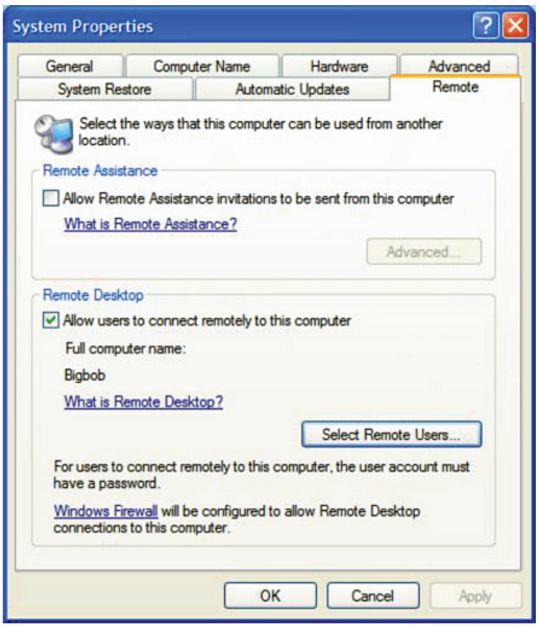


FIGURE 7-17. REMOTE TAB

- Check the Allow users to connect remotely to this computer checkbox.
- Click the Select Remote Users… button.

  The Remote Desktop Users window opens.

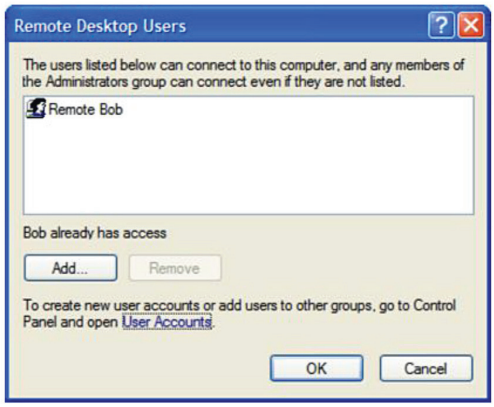


FIGURE 7-18. REMOTE DESKTOP USERS SCREEN

- Click the Add… button to add users to the list of those allowed to remotely access the system using the RDP protocol.
- Click OK to close the Remote Desktop Users window.
- Click OK to close the System Properties window.

Windows generates the available user list from local accounts on the target Windows computer. To setup new users to then add them to the Remote Desktop Users list:

- Navigate to Start Menu > Control Panel.

- Double-click the User Accounts icon.
- Create new users as required.

NOTE: When a remote user connects to the accessed computer via the root console, Remote Desktop automatically locks that computer (so no other user can access the applications and files). When you come back to your computer, you can unlock it by typing CTRL+ALT+DEL.

7.8.2 CONFIGURE THE REMOTE DESKTOP CONNECTION CLIENT

Once the Client computer is securely connected to the console server (either locally, or remotely through an enterprise VPN, a secure SSH internet tunnel or a dial-in SSH tunnel), you can establish the Remote Desktop connection from the Client. To do this, enable the Remote Desktop Connection on the remote client PC then point it to the SDT Secure Tunnel port in the console server.

On a Windows client:

- Navigate to Start Menu > Programs > Accessories > Communications.
- Click Remote Desktop Connection.
- Enter the appropriate IP address and port number in Computer.

  Where there is a local connection or enterprise VPN connection, enter the IP Address of the console server, and the port number of the SDT Secure Tunnel for the console server serial port that is attached to the Windows computer to be controlled.

  For example, if the Windows computer is connected to serial Port 3 on a console server located at 192.168.0.50, enter 192.168.0.50:7303.

  Where there is an SSH tunnel over a dial up PPP connection or over a public internet connection or private network connection, enter localhost as the IP address (that is, 127.0.0.1). For Port Number, enter the source port you created when setting up SSH tunneling/port forwarding (see Section 7.1).
- Click Option.
- Specify an appropriate color depth in the Display section.

  For example, for a connection running over a modem, don't set the color depth to greater than 256 colors (8-bit).
- In Local Resources, specify the peripherals and ports on the remote Windows computer that are available to be controlled (for example, a directly connected printer or the serial port on the Windows PC).

FIGURE 7-19. REMOTE DESKTOP CONNECTION SCREEN

- Click Connect.

On a Linux or UNIX client:

- Launch the open source rdesktop client from a shell. For example:

  rdesktop -u windows-user-id -p windows-password -g 1200x950 ms-windows-terminal-server-host-name

TABLE 7-3. RDESKTOP OPTIONS

RDESKTOP OPTION	DESCRIPTION
-a	Color depth. Valid values are 8, 16, and 24.
-r	Device redirection. Redirects remote machine sound to the local device.
-g	Display geometry. Either widthxheight in pixels, or % of local screen.
-p	Sets rdesktop to receive a password prompt form the remote machine.

You can use GUI front end tools such as the GNOME Terminal Services Client tsclient to configure and launch the rdesktop client.

tsclient also allows for multiple rdesktop configurations for connection to many servers.

On an OS X client

- Download Microsoft's free Remote Desktop Connection (RDC) client from https://microsoft.com/en-us/download/details.aspx?id=18140.

NOTE: Microsoft RDC Client for OS X is not supported for use with OS X v10.7 (Lion) or later.

7.9 SDT SSH TUNNEL FOR VNC

Alternately, with SDT and Virtual Network Computing (VNC), Users and Administrators can securely access and control computers running Windows, Linux, macOS, Solaris and UNIX.

To set up a secure VNC connection you must

- Install (if necessary) and configure VNC Server software on the computer to be accessed.
- Install (if necessary) and configure VNC Viewer software on the Viewer PC.

7.9.1 INSTALL AND CONFIGURE THE VNC SERVER ON THE COMPUTER TO BE ACCESSED

Virtual Network Computing (VNC) software enables users to remotely access computers running Linux, macOS, Solaris, UNIX, all versions of Windows and most other operating systems.

VNC Servers

RealVNC Connect, https://realvnc.com/, is a multi-platform VNC server that runs on Windows, macOS, Linux, Solaris, HP-UX, AIX, and Raspberry Pi. RealVNC also offers a VNC client, RealVNC Viewer, which runs on these platforms as well as iOS, Android, and Chrome.

TightVNC, https://tightvnc.com/, is a dual-licensed (GPL and commercial) VNC server for Windows. TightVNC also offer a Java-based VNC viewer. It works on any system with Java SE version 1.6 or later installed.

UltraVNC, http://uvnc.com/, is a VNC server and viewer for Windows.

Most Linux distributions ship with VNC servers and viewers. If a Linux instance does not have VNC software installed, it will likely be available for install via the distro's software repository.

For example, to turn the VNC server on in Centos 7:

- Navigate to Applications > System Tools > Settings.
- Click Sharing.
- Click Screen Sharing.
- Click the On-Off control to start the VNC server.

Below the Screen Sharing title bar is the vnc-protocol URL the computer is accessible via.

- Click the Require a password radio button.
- Create and Enter the Password remote clients must enter to view the Centos screen.
- Click the Close box in the top right-hand corner of the Screen Sharing window.
- Click the Close box in the top right-hand corner of the Sharing window.

macOS also ships with a VNC server. To turn this server on:

- Choose Apple Menu > System Preferences.
- Click the Sharing icon (or choose View > Sharing).
- Check the Screen Sharing checkbox.

The built-in VNC server is now running. Immediately below the text Screen Sharing: On is the vnc-protocol URL the computer is accessible via.

- Click the Computer Settings… button.
- Check the VNC viewers may control screen with password checkbox.
- Create and enter the password said VNC viewer applications will need to supply.
- Click OK.

7.9.2 INSTALL, CONFIGURE AND CONNECT THE VNC VIEWER

VNC is platform-independent: a viewer on one OS can connect to a server on any other OS.

There are also Java viewers available so that any desktop can be viewed with any Java-capable browser. http://en.wikipedia.org/wiki/VNC lists many VNC viewers sources.

To make VNC faster, when you set up the VNC viewer:

- If you have a fast enough CPU, set encoding to ZRLE.
- Decrease the color level (for example, 64-bit).
- Disable the background transmission on the server, or use a plain wallpaper.

See http://doc.uvnc.com/ for detailed configuration instructions.

To establish a VNC connection:

- Enter the VNC server IP address and port.

  When the viewer is connected via an SSH tunnel, whether over the Internet, a dial-in connection, or a private network, use localhost or 127.0.0.1 as the VNC server's IP address.

  The port number is the number entered when setting up SSH tunneling/port forwarding in Section 7.2.6. For example: 1234.

FIGURE 7-20. VNC SERVER SCREEN 1

When the VNC viewer is connected directly to the console server (that is locally or remotely through a VPN or dial in connection) and the VNC server is serially connected to the console server, enter the IP address of the console server unit with the TCP port that the SDT tunnel will use.

The TCP port will be 7900 plus the physical serial port number (that is 7901 to 7948). All traffic directed to port 79xx on the console server is tunneled thru to port 5900 on the PPP connection on serial Port xx.

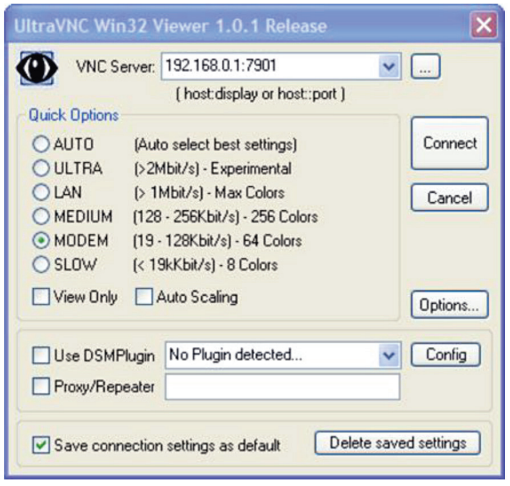


FIGURE 7-21. VNC SERVER SCREEN 2

For example, for a Windows computer using UltraVNC as the viewer connecting to a VNC server which is attached to Port 1 on a console server located 192.168.0.1:

- Establish the VNC connection by activating the VNC viewer and entering the Password.

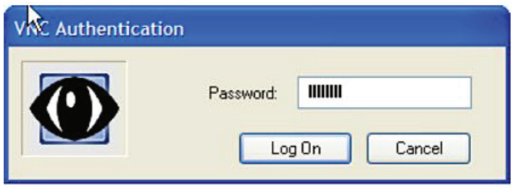


FIGURE 7-22. ENTER PASSWORD

For background reading on Remote Desktop and VNC access, we recommend the following:

- The Microsoft Remote Desktop How-To: http://www.microsoft.com/windowsxp/using/mobility/getstarted/remoteintro.mspx.
- The Illustrated Network Remote Desktop help page: http://theillustratednetwork.mvps.org/RemoteDesktop/RemoteDesktopSetupandTroubleshooting.html.
- What is Remote Desktop in Windows XP and Windows Server 2003? by Daniel Petri: http://www.petri.co.il/what's_remote_desktop.htm.
- Frequently Asked Questions about Remote Desktop: http://www.microsoft.com/windowsxp/using/mobility/rdfaq.mspx.
- Secure remote access of a home network using SSH, Remote Desktop and VNC for the home user: http://theillustratednetwork.mvps.org/RemoteDesktop/SSH-RDP-VNC/RemoteDesktopVNCandSSH.html.
- Wikipedia's general background article on VNC: http://en.wikipedia.org/wiki/VNC.

7.10 USING SDT TO IP CONNECT TO HOSTS THAT ARE SERIALLY-ATTACHED TO THE GATEWAY

Network (IP) protocols like RDP, VNC and HTTP can also be used for connecting to host devices that are serially connected through their COM port to the console server. To do this you must:

- Establish a PPP connection between the host and the gateway. See Section 7.10.1.
- Set up Secure Tunneling Ports on the console server. See Section 7.10.2.
- Configure SDT Connector to use the appropriate network protocol to access IP consoles on the host devices that are attached to the console server serial ports. See Section 7.10.3.

7.10.1 ESTABLISH A PPP CONNECTION BETWEEN THE HOST COM PORT AND CONSOLE SERVER

- This step is only necessary for serially-connected computers. Physically connect the COM port on the host computer that is to be accessed, to the serial port on the console server.
- On computers running Linux, UNIX, Solaris and other Unix-like operating systems, establish a PPP connection over the serial port.
- The online tutorial at http://yolinux.com/TUTORIALS/LinuxTutorialPPP.html presents a selection of methods for establishing a PPP connection using a computer running Linux.
- On computers running Windows, follow the procedure below to set up an advanced network connection between the Windows computer's COM port and the console server.
- Windows allows for the creation of a simple dial-in service which can be used for a Remote Desktop or VNX or HTTP/X connection to the console server.
- Navigate to Start Menu > Control Panel.
- Double-click the Network Connections icon.
- Click the New Connection Wizard.

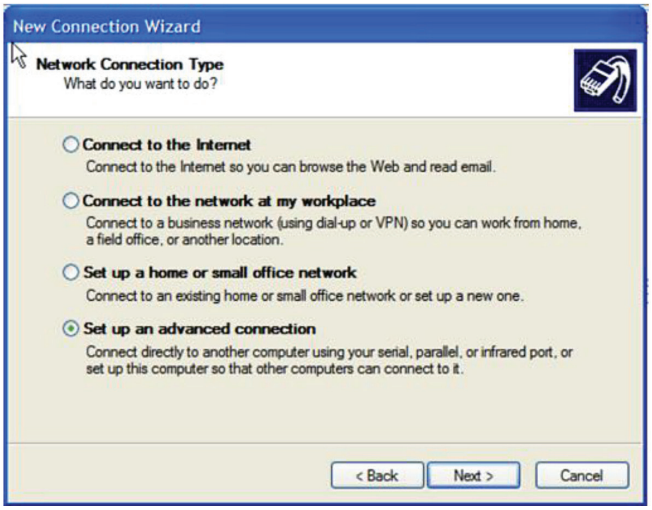


FIGURE 7-23. NEW CONNECTION WIZARD SCREEN

- Select the Set up an advanced connection radio button.
- Select Accept Incoming Connections in the Advanced Connection Options window.
- click Next.
- Select COM1 as the Connection Device (that is, the COM port on the computer that is connected to the console server's serial port).
- Set the COM port to its maximum baud rate.
- Click Next.
- Select Do not allow virtual private connections in the Incoming VPN Connection Options window.
- Click Next.
- Select which users will be allowed to use this connection.

  This should be the same users given Remote Desktop access privileges in the earlier step.
- Click Next.
- Select TCP/IP in the Network Connections window.
- Click Properties.
- Select Specify TCP/IP addresses in the Incoming TCP/IP Properties window.
- Enter IP addresses in the From and To fields.

  Choose any TCP/IP addresses so long as they are addresses that are not used anywhere else on your network.

  The From address will be assigned to the computer running Windows. The To address will be used by the console server. For simplicity, use the IP address shown next:

  From 169.134.13.1

  To 169.134.13.2

  Alternatively, set the advanced connection and access on the computer running Windows to use the console server defaults:

  From 10.233.111.254

  Allow calling computer to specify its own IP address checked.
- Click OK.

Another option is to use the console server's default username and password to setup the Remote Desktop user and give this user permission to use the advanced connection to access the computer running Windows.

- The console server's default Username is portXX where XX is the serial port number on the console server.
- The console server's default Password is portXX where XX is the serial port number on the console server.

  For example, for an RDP connection to serial port 2 on the console server, set up a Windows user named port02 with appropriate permissions.

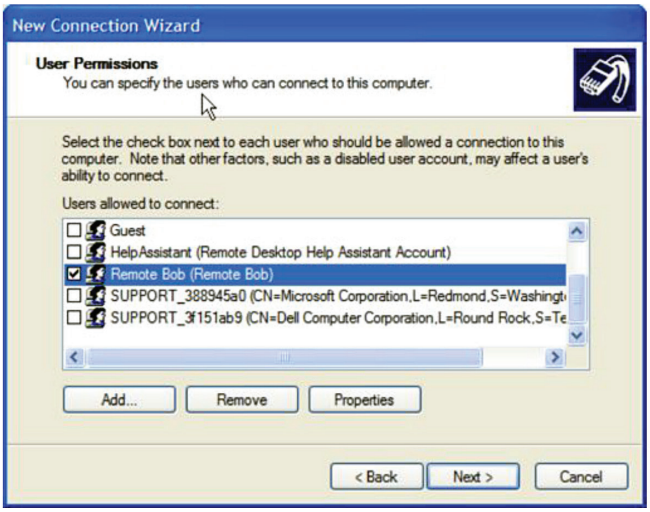


FIGURE 7-24. USER PERMISSIONS SCREEN

When the PPP connection has been set up, a network icon appears in the Windows task bar.

The above notes describe setting up an incoming connection for Windows XP. The steps are similar for later versions of Windows although the setup screens present slightly differently.

If an Incoming Connections Properties window presents, check the Always allow directly connected devices such as palmtop computers to connect without providing a password checkbox.

Also the option to Set up an advanced connection is not available in Windows 2003 if RRAS is configured. If RRAS has been configured, enable the null modem connection for the dial-in configuration.

7.10.2 SET UP SDT SERIAL PORTS ON THE CONSOLE SERVER

To set up RDP (and VNC) forwarding on the console server Serial Port that is connected to the Windows computer COM port:

- Navigate to Serial & Network > Serial Port.
- Click Edit for the particular Serial Port connected to the Windows computer's COM port.
- Select SDT Mode in the SDT Settings section.

  This will enable port forwarding and SSH tunneling.

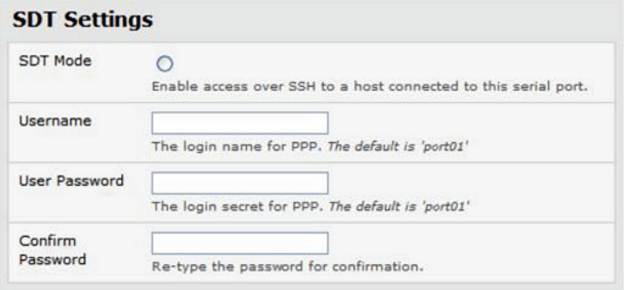


FIGURE 7-25. SDT SETTINGS SCREEN

NOTE: Enabling SDT overrides all other configuration protocols on this port.

- Enter a Username and User Password.

  If you leave the Username and User Password fields blank, they both default to portXX where XX is the serial port number. For example, the default username and password for Secure RDP over Port 2 is port02.

- Set the console server's serial port Common Settings (Baud Rate and Flow Control) to the same values as were set up on the Windows computer's COM port.
- Click Apply.

RDP and VNC forwarding over serial ports is enabled on a per-Port basis. You can add Users who can have access to these ports (or reconfigure User profiles) by navigating to Serial & Network > User & Groups as documented in Chapter 5.

7.10.3 SET UP SDT CONNECTOR TO SSH PORT FORWARD OVER THE CONSOLE PORT

In the SDT Connector software running on your remote computer, specify the gateway IP address of your console server and specify a username and password for a user you have set up on the console server that has access to the desired port.

Next add a New SDT Host.

In the Host address put portxx where xx is the port you are connecting to.

For example, for port 3 enter a Host Address of port03 and then check the RDP Service check box.

7.11 SSH TUNNELING USING OTHER SSH CLIENTS (FOR EXAMPLE, PUTTY)

SDT Connector, which is supplied with console servers, is Black Box's recommended SSH client. There are other SSH client programs that can provide secure SSH connections to console servers and connected devices.

TABLE 7-4. SSH CLIENTS

SSH CLIENT	SOURCE	DESCRIPTION
PuTTY	http://putty.org/	An open-source SSH implementation for Windows.
SSHTerm	http://sourceforge.net/projects/sshtools	A Java-based open-source SSH communications suite.
Tectia SSH	https://ssh.com/products/tectia-ssh/	A commercial SSH client and server.
Reflection for Secure IT	https://www.microfocus.com/products/reflection-secure-it	A commercial SSH client and server.

This section documents the use of the PuTTY client to establish an SSH-tunneled connection to a network-connected device.

- Launch PuTTY.

  The PuTTY Configuration window opens.
- Click Session in the Category section.
- Enter the IP address of the console server to connect to in the Host Name or IP address field.

  For dial-in connections, this IP address will be the Local Address that you assigned to the console server when you set it up as the Dial-In PPP Server.

  For Internet or local/VPN connections, this will be the public IP address of the console server.
- Leave the port number as 22 (unless you've configured the console server to run SSH on a port other than the default value of 22).
- Click the SSH radio button under Connection type.
- Click Tunnels in the Category section (in the disclosure tree this is in Connection > SSH).
- Enter any high, unused port number (for example: 55555) in the Source port field under Add new forwarded port.
- Enter the Destination IP address and port.

  If your destination device is network connected to the console server and you are connecting using RDP, set the Destination as:

  managed-device-ipaddress-or-hostname:3389

  For example, if, when setting up the Managed Device as Network Host on the console server its IP address was set to:

  192.168.253.1

  or its hostname was set to:

  accounts.myco.intranet.com

  then set the Destination to:

  192.168.523.1:3389

  or

  accounts.myco.intranet.com:3389

NOTE: Only devices that have been configured as networked Hosts can be accessed using SSH tunneling (except by the root user who can tunnel to any IP address the console server can route to).

If your destination computer is serially-connected to the console server, set the Destination as:

port-label:3389

For example, if the Label you specified on the serial port on the console server is win2k3, then specify the remote host as:

win2k3:3389

Alternatively, set the Destination as:

portXX:3389

where XX is the SDT-enabled serial port number.

For example, if port 4 on the console server carries the RDP traffic then set the Destination to:

port04:3389

- Select the Local radio button.
- Click Add.
- Click Open.

  A shell prompt window will open, prompting you to login as:
- Enter a username and press Return.

  The shell will return a password prompt.
- Enter the user's password. and press Return.

  If you are connecting as a user in the users group you can only SSH tunnel to hosts and serial ports where you have specific access permissions.

  If you are connecting as an administrator (that is, a user in the admin group), then you can connect to any configured host or serial port that has SDT enabled.

To set up the secure SSH tunnel for a HTTP browser connection to the managed device, specify port 80 (rather than port 3389, used for RDP) in the Destination IP field.

To set up the secure SSH tunnel from the Client PC to the console server for VNC, configure the VNC port redirection by specifying port 5900 in the Destination IP field.

7.12 VNC SECURITY

How secure is VNC? VNC access generally allows access to your whole computer, so security is very important. VNC uses a random challenge-response system to provide the basic authentication that allows you to connect to a VNC server. This is reasonably secure and the password is not sent over the network.

Once connected, all subsequent VNC traffic is unencrypted. So a malicious user could snoop your VNC session. Also there are VNC scanning programs available that will scan a subnet looking for PCs which are listening on one of the ports which VNC uses.

Tunneling VNC over a SSH connection ensures all traffic is strongly encrypted. No VNC port is ever open to the internet, so anyone scanning for open VNC ports will not be able to find your computers. When tunneling VNC over a SSH connection, the only port that you are opening on your console server is the SDT port (port 22).

It may be prudent to tunnel VNC through SSH even when the Viewer PC and the console server are both on the same local network.

CHAPTER 8: ALERTS, AUTO-RESPONSE AND LOGGING

This chapter describes the automated response, alert generation and logging features of the console server.

The Auto-Response facility extends on the basic Alert facility available in earlier (pre V3.5) firmware revisions. With Auto-Response, the console server monitors selected serial ports, logins, the power status and environmental monitors and probes for Check Condition triggers. The console server will then initiate a sequence of actions in response to these triggers. To configure Auto-Response:

- Set the general parameters.
- Select and configure the Check Conditions (the conditions that trigger the response).
- Specify the Trigger Actions (the action sequence initiated in case of the trigger condition).
- Specify the Resolve Actions (the actions performed when trigger conditions are resolved).

All console server models can maintain log records of all access and communications with the console server and with the attached serial devices. A log of all system activity is also maintained as is a history of the status of any attached environmental monitors.

Some models can also log access and communications with network attached hosts and maintain a history of the UPS and PDU power status.

If port logs are to be maintained on a remote server, then the access path to this location needs to be configured Then you need to activate and set the desired levels of logging for each serial and network port and for power and environment UPS (see Chapter 9).

8.1 CONFIGURE AUTO-RESPONSE

With the Auto-Response facility, a sequence of Trigger Actions is initiated in case of a specified trigger condition (the Check Condition). Subsequent Resolve Actions can also be performed when the trigger condition has been resolved.

First, set the general parameters that will be applied to all Auto-Responses.

- Navigate to Alerts & Logging > Auto-Response.
- Check the Log Events checkbox in the Global Auto-Response Settings section. This enables logging of all Auto-Response activities.

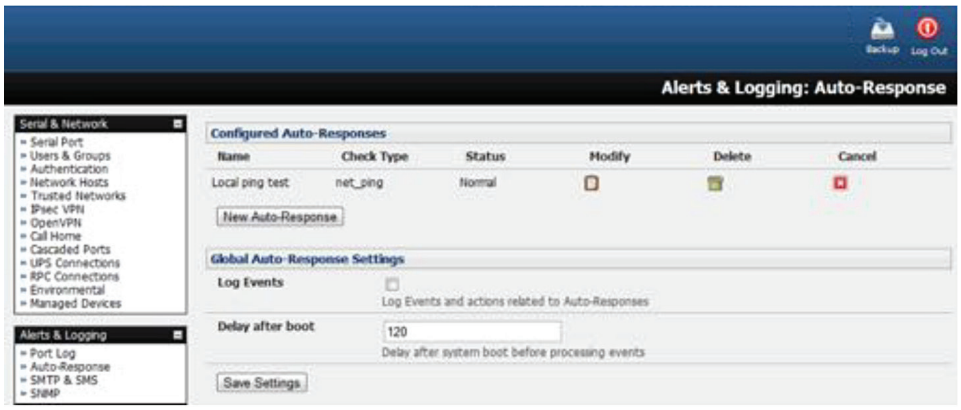


FIGURE 8-1. CONFIGURED AUTO RESPONSES SCREEN

- Set the Delay after boot time (in seconds) to establish the delay between a console server booting and the same console server processing events.

To configure a new Auto-Response:

- Select New Auto-Response in the Configured Auto-Response field.

  A new Auto-Response Settings page presents.

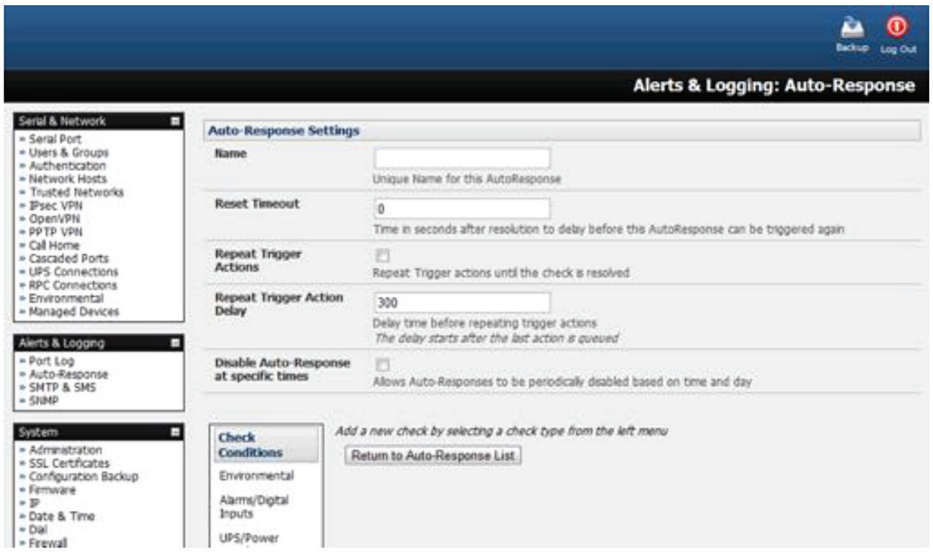


FIGURE 8-2. AUTO RESPONSE SETTINGS PAGE

- Enter a unique Name for the new Auto-Response.
- Specify the Reset Timeout for the time in seconds after resolution to delay before this Auto-Response can be triggered again.
- Check Repeat Trigger Actions to repeat trigger actions until the check is resolved.
- Enter any required delay time before repeating trigger actions in Repeat Trigger Action Delay. This delay starts after the last action is queued.
- Check Disable Auto-Response at specific times and you will be able to periodically disable auto-Responses between specified times of day.

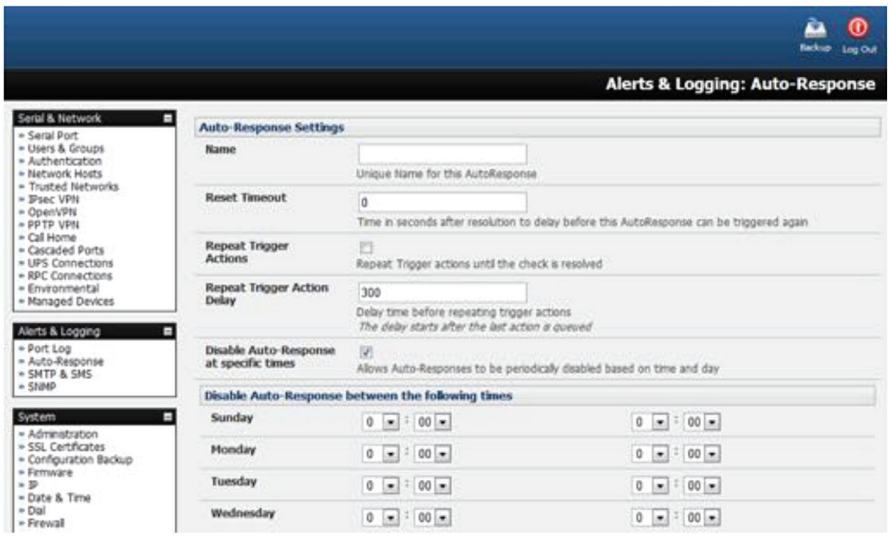


FIGURE 8-3. DISABLE AUTO RESPONSE SETTINGS PAGE

8.2 CHECK CONDITIONS

To configure the condition that will trigger the Auto-Response:

- Click on the Check Condition type (for example, Environmental, UPS Status or ICMP ping) to be configured as the trigger for this Auto-Response in the Auto-Response Settings menu.

8.2.1 ENVIRONMENTAL

To configure Humidity or Temperature levels as the trigger event:

- Click on Environmental as the Check Condition.
- In the Environmental Check menu, select the specific environmental sensor to be checked for the trigger.

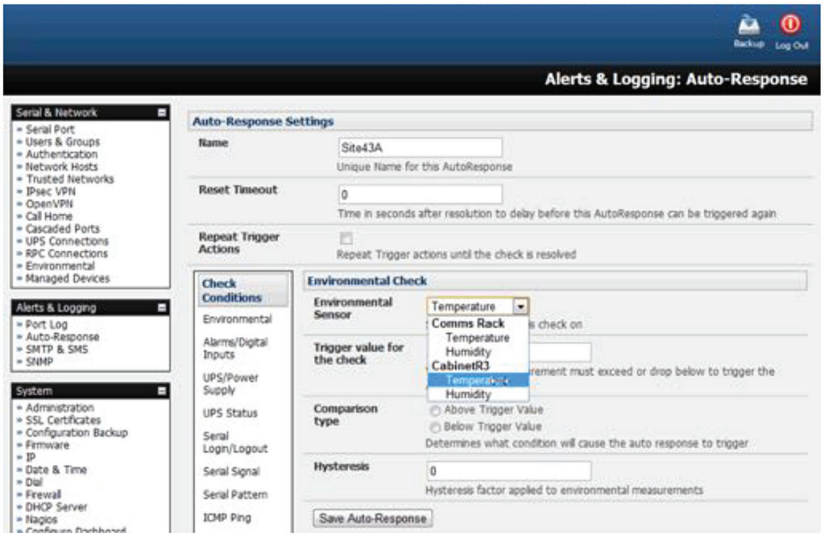


FIGURE 8-4. ENVIRONMENTAL CHECK MENU, TEMPERATURE SELECTED

- Specify the Trigger value (in °C or °F for temperature and % for humidity) that the check measurement must exceed or drop below to trigger the AutoResponse.
- Select Comparison type as being Above Trigger Value or Below Trigger Value to trigger.
- Specify any Hysteresis factor that is to be applied to environmental measurements. For example, if an Auto-Response was set up with a trigger event of a temp reading above 49°C with a Hysteresis of 4, then the trigger condition would not be seen as having been resolved till the temperature reading was below 45°C.
- Check Save Auto-Response.

NOTE: Before configuring Environmental Checks as the trigger in Auto-Response, you will need first to configure the Temperature sensors, the Humidity sensors or both on your LES1200 or attached EMD.

8.2.2 ALARMS AND DIGITAL INPUTS

To set the status of any attached Smoke or Water sensors or digital inputs as the trigger event:

- Click on Alarms/Digital Inputs as the Check Condition.
- In the Alarms/Digital Inputs Check menu, select the specific Alarm/Digital IO Pin that will trigger the Auto-Response.
- Select Trigger on Change to trigger when alarm signal changes, or select to trigger when the alarm signal state changes to either a Trigger Value of Open (0) or Closed (1).
- Check Save Auto-Response.

NOTE: Before configuring Alarms/Digital Inputs checks in Auto-Response you first must configure the sensor/DIO that is to be attached to your EMD or LES1200.

8.2.3 UPS AND POWER SUPPLY

To use the properties of any attached UPS as the trigger event:

- Select UPS/Power Supply as the Check Condition.
- Select the UPS Power Device Property (Input Voltage, Battery Charge %, Load %, Input Frequency Hz or Temperature in °C) to be checked for the trigger.
- Specify the Trigger value that the check measurement must exceed or drop below to trigger the AutoResponse.
- Select the Comparison type as being Above Trigger Value or Below Trigger Value to trigger.
- Specify any Hysteresis factor to be applied to environmental measurements.

For example, if an Auto-Response is set up with a trigger event of a battery charge below 20% with a Hysteresis of 5 then the trigger condition will not be set to resolved until the battery charge is above 25%.

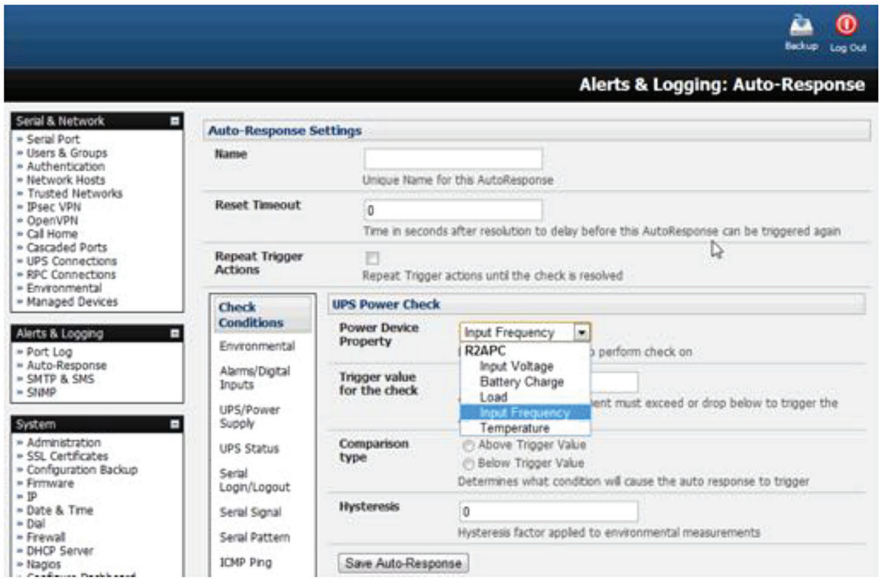


FIGURE 8-5. UPS AND POWER SUPPLY SCREEN

- Check Save Auto-Response.

NOTE: Before configuring UPS checks in Auto-Response you first must configure the attached UPS.

8.2.4 UPS STATUS

To use the alert state of any attached UPS as the Auto-Response trigger event:

- Click on UPS Status as the Check Condition.
- Select the reported UPS State to trigger the Auto-Response (either On Battery or Low Battery).

The Auto-Response will resolve when the UPS state returns to the Online state.

- Select which connected UPS Device to monitor.
- Click Save Auto-Response.

NOTE: Before configuring UPS state checks in Auto-Response, the attached UPS must be configured.

8.2.5 SERIAL LOG-IN, SIGNAL OR PATTERN

To monitor serial ports and check for login/logout or pattern matches for Auto-Response triggers events:

- Click on Serial Login/Logout as the Check Condition.
- In the Serial Login/Logout Check menu, select Trigger on Login (to trigger when any user logs into the serial port) or Trigger on Logout.
- Specify Serial Port to perform check on.
- Click on Serial Signal as the Check Condition.

  The above two options can be set individually or together.
- In the Serial Signal Check menu, select the Signal (CTS, DCD, DSR) to trigger the condition (either on serial signal change, or check level).
- Specify Serial Port to perform check on.
- Click on Serial Pattern as the Check Condition.

  The above two options can be set individually or together.
- In the Serial Pattern Check menu, select the PCRE pattern to trigger on and the serial line (TX or RX) and Serial Port to pattern check on.

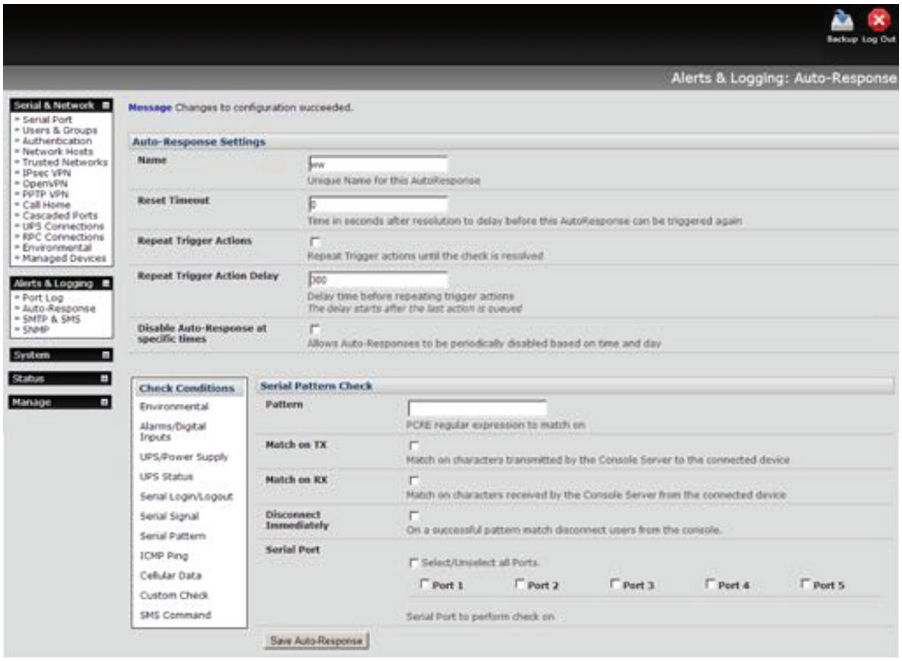


FIGURE 8-6.

NOTE: With Serial Pattern checks, you can nominate to Disconnect Immediately all users from the serial port being monitored in the event of a successful pattern match.

NOTE: For devices with a built-in cellular modem with GPS enabled, the GPS will be displayed as an additional port and it can be monitored for trigger events. For example, with a console server with 4 serial ports, the GPS will be shown as Port 5.

- Click Save Auto-Response.

NOTE: Before configuring serial port checks in Auto-Response, the affected serial port must be configured in Console server mode. Most serial port checks are not resolvable so resolve actions will not run.

8.2.6 USB CONSOLE STATUS

NOTE: USB port labels in the Web interface match the USB port labels printed on a console server with two exceptions. Some console servers include discrete pairs of USB ports that do not have printed labels. In this case, the Web interface denotes them as either Upper or Lower. The Web interface lists them by their physical relationship to each other. Some console servers ship with an array of four USB ports. A limited number of these console servers have labels A–D printed by these ports even though the Web interface will denote them as USB ports 1–4.

To monitor USB ports:

- Select USB Console Status as the Check Condition.
- Check the Trigger on Connect checkbox, the Trigger on Disconnect checkbox, or both checkboxes to set which actions trigger the Auto-Response.
- Check each USB port to be monitored (or click the Select/Unselect all Ports checkbox to select or deselect all USB ports).
- Click the Save Auto-Response button.
- Select an option from the Add Trigger Action list.
- Enter a unique Action Name for the trigger action being created.

- Set an Action Delay Time.

  By default, this is 0 seconds.
- Enter the specific details of the selected action. For example, the Send Email action requires a Recipient Email Address and allows for a Subject and Email Text.
- Click the Save New Action button.

NOTE: USB console status checks are not resolvable. Trigger actions run but Resolve actions do not.

8.2.7 ICMP PING

To use a ping result as the Auto-Response trigger event:

- Select ICMP Ping as the Check Condition.
- Specify which Address to Ping (that is, the IP address or DNS name to send ICMP pings to).
- Specify which Interface to send ICMP pings from (for example, the Management LAN or Wireless network).
- Set the Check Frequency. This is the time in seconds between checks.
- Set the Number of ICMP Ping packets to send.
- Check Save Auto-Response.

8.2.8 CELLULAR DATA

This check monitors the aggregate data traffic inbound and outbound through the cellular modem as an Auto-Response trigger event.

- Select Cellular Data as the Check Condition.

NOTE: Before configuring cellular data checks in Auto-Response, the internal cellular modem must be configured and detected by the console server.

8.2.9 CUSTOM CHECK

This check allows users to run a nominated custom script with nominated arguments whose return value is used as an Auto-Response trigger event:

- Click on Custom Check as the Check Condition.
- Create an executable trigger check script file.

  For example /etc/config/test.sh

```
#!/bin/sh
logger "A test script"
logger Argument1 = $1
logger Argument2 = $2
logger Argument3 = $3
logger Argument4 = $4
if [ -f /etc/config/customscript.0 ]; then
```

```
    rm /etc/config/customscript.0
    exit 7
fi
touch /etc/config/customscript.0
exit 1
```

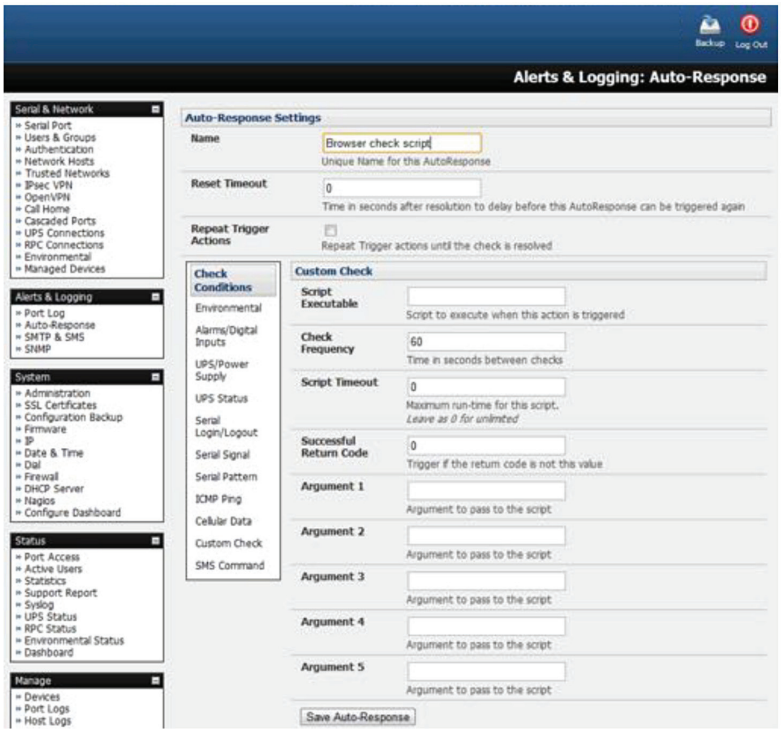


FIGURE 8-7.

- Enter the Script Executable file name. For example /etc/config/test.sh.
- Set the Check Frequency. This is the time, in seconds, between re-running the script.
- Set the Script Timeout. This is the maximum run-time for the script.
- Specify the Successful Return Code. An Auto-Response is triggered if the return code from the script is not this value.
- Enter Arguments that are to be passed to the script. For example, with a web page html check script, these Arguments might specify the web page address/DNS and user logins.
- Check Save Auto-Response.

8.2.10 CLI SESSION EVENT

When the Check Condition is set to CLI Session Event, the triggers that cause the Auto-Response to run can be any or all of the following:

- Trigger on Login
- Trigger on Logout
- Trigger on Authentication Error

Checking the Trigger on Login checkbox sets the Auto-Response to run when the console's shell is logged in to.

Checking the Trigger on Logout checkbox sets the Auto-Response to run when the console's shell is logged out of.

And checking the Trigger on Authentication Error checkbox sets the Auto-Response to run when the console's shell returns an authentication error.

An Auto-Response can be set to trigger on one, two or all three of these events. After selecting the desired CLI session events to respond to:

- Click Save Auto-Response.

8.2.11 SMS COMMAND

An incoming SMS command from a nominated caller can trigger an Auto-Response:

- Select SMS Command as the Check Condition.

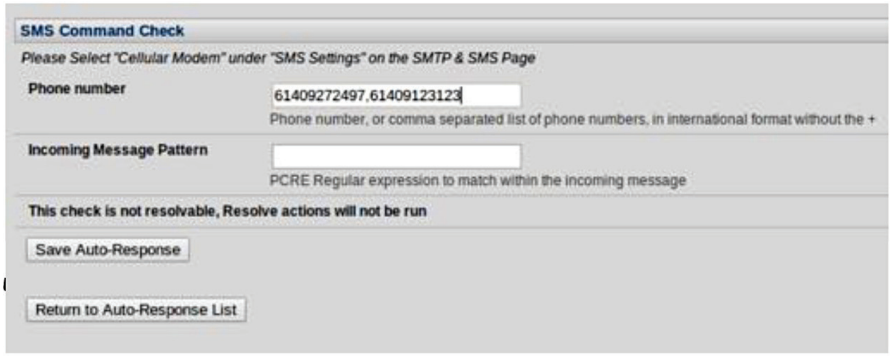


FIGURE 8-8.

- Set the Phone number. For multiple SMS sources comma-separate the numbers.

NOTE: Enter the phone number in international format without the plus-sign (+) prefix.

- Set the Incoming Message Pattern to match to create trigger event. This pattern is a PCRE regular expression.

NOTE: The SMS command trigger condition can only be set if an internal cellular modem is detected.

- Click Save Auto-Response.

8.2.12 LOGIN AND LOGOUT CHECK

To configure Web Log In/Out as the trigger event:

- Select Web UI Authentication as the Check Condition.
- Check Trigger on Login or Trigger on Logout to trigger if a user logs into or out of the Web UI.

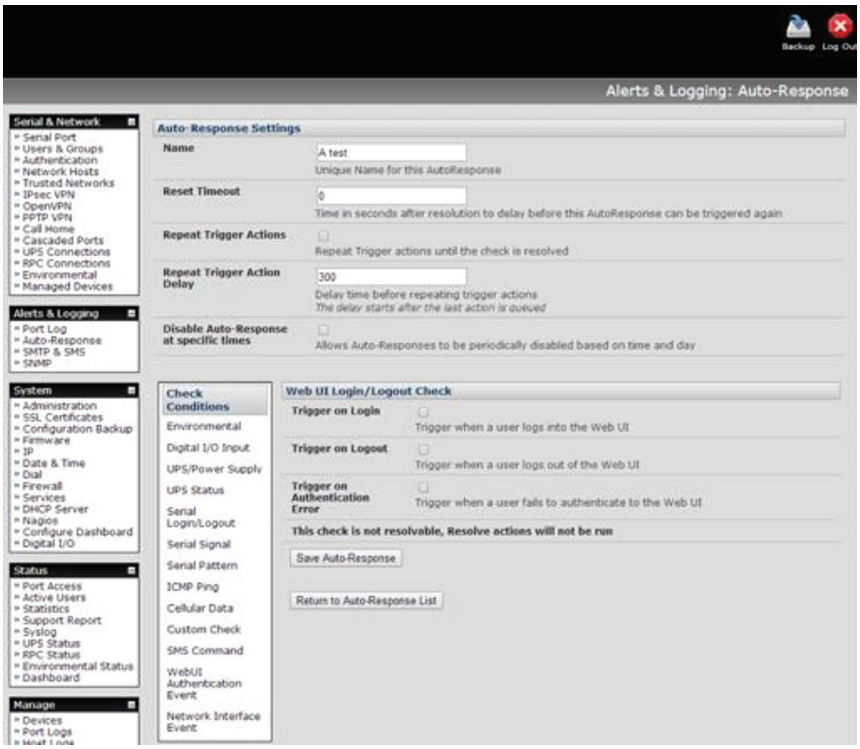


FIGURE 8-9.

- Check Trigger on Authentication Error to trigger when Web UI user authentication fails.

NOTE: This check is not resolvable. Resolve actions will not, as a consequence, run.

- Click Save Auto-Response.

8.2.13 NETWORK INTERFACE EVENT

You may wish to configure a change in the network status as the trigger event (e.g., to send an alert or restart a VPN tunnel connection):

- Select Network Interface as the Check Condition.
- Select the Interface to monitor.
- Check the interface Event to trigger on.

NOTE: This check is not resolvable. Resolve actions will not, as a consequence, run.

- Click Save Auto-Response.

8.2.14 ROUTED DATA USAGE CHECK

This check monitors the specified input interface for data usage that is being routed through the console server and out another interface such as the internal cellular modem.

It is particularly useful in IP Passthrough mode to detect when the downstream router has failed over and is now routing via the modem as a backup connection.

This check may be configured with these parameters:

- The Black Box's incoming Interface to monitor.
- An optional Source MAC address or source IP Address, to monitor traffic from a specific host (for example, the downstream router).
- A Data Limit threshold and specified Time Period.

The Auto-Response will trigger when the limit is reached in the specified time.

The Auto-Response will resolve if no matching data is routed for the Resolve Period.

8.3 TRIGGER ACTIONS

To configure the sequence of actions that is to be taken in the event of the trigger condition:

- For a nominated Auto-Response with a defined Check Condition, select an Add Trigger Action (for example, Send Email or Run Custom Script).

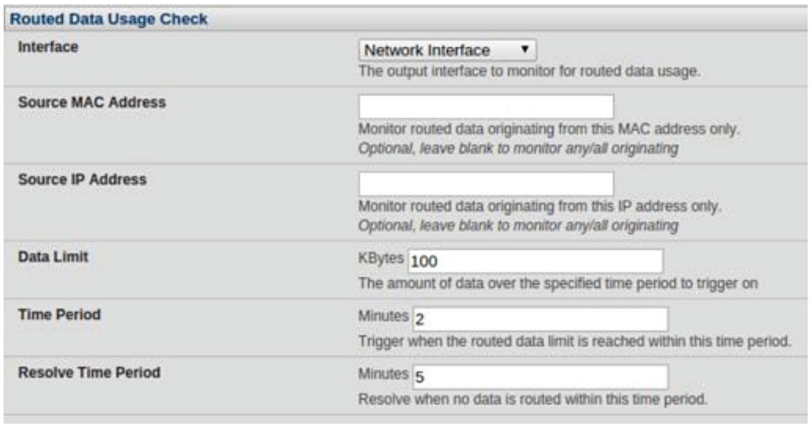


FIGURE 8-10.

This selects the action type to be taken.

- Configure the selected action (as detailed in the sections following).

Each action is configured with a nominated Action Delay Time which specifies how long (in seconds) after the Auto-Response trigger event to wait before performing the action.

NOTE: You can add follow-on actions to create a sequence of actions that will be taken in the event of the one trigger condition.

To edit or delete an existing action:

- Click the Modify or Delete icon in the Scheduled Trigger Action table.

A message text can be sent with Email, SMS and Nagios actions. This configurable message can include selected values:

TABLE 8-1. MESSAGE TEXT

VALUE	DESCRIPTION
$AR_TRIGGER_VAL	The trigger value for the check. For example the UPS Status trigger value can be either onbatt or battlow..
$AR_VAL	The value returned by the check. For example the UPS status value can be online, onbatt, or battlow..
$AR_CHECK_DEV	The name of the device being checked. For example, for Alarm, the alarm name.
$TIMESTAMP	The current timestamp.
$HOSTNAME	The hostname of the console server.

The default message text is:

$TIMESTAMP: This action was run — Check details: value $AR_VAL vs trigger value $AR_TRIGGER_VAL.

8.3.1 SEND E-MAIL

- Select Send Email as the Add Trigger Action.
- Enter a unique Action Name.
- Set the Action Delay Time.
- Specify the Recipient Email Address to send this email to. For multiple recipients enter comma-separated addresses.
- Enter a Subject for the email.
- Edit the Email Text message to send.
- Click Save New Action.

NOTE: An SMS alert can also be sent via an SMTP (email) gateway. You will need to specify the Recipient Email Address in the format specified by the gateway provider. For example, for T-Mobile it is phonenumber@tmomail.net.

8.3.2 SEND E-MAIL

- Select Send SMS as the Add Trigger Action.
- Enter a unique Action Name.
- Set the Action Delay Time.
- Specify the Phone number that the SMS will be sent to. This must be in international format bit without the leading plus (+) sign.
- Edit the Message Text to send
- Click Save New Action.

NOTE: The SMS alert can be sent if there is an internal cellular modem attached. Alternatively, an SMS alert can also be sent via a SMTP SMS gateway as documented in Section 8.5.2.

8.3.3 PERFORM RPC ACTION

- Select Perform RPC Action as the Add Trigger Action.
- Enter a unique Action Name.
- Set the Action Delay Time.
- Select a power Outlet.
- Specify the Action (Power On, Power Off, or Cycle) to be performed.
- Click Save New Action.

8.3.4 RUN CUSTOM SCRIPT

- Select Run Custom Script as the Add Trigger Action.
- Enter a unique Action Name.
- Set the Action Delay Time.
- Create a script file to execute when this action is triggered.
- Enter the Script Executable's file name. For example /etc/config/action.sh.
- Set the Script Timeout. This is the maximum run-time for the script. Set this at 0 for unlimited time.
- Enter any Arguments that are to be passed to the script.
- Click Save New Action.

8.3.5 SEND SNMP TRAP

- Select Send SNMP Trap as the Add Trigger Action.
- Enter a unique Action Name.
- Set the Action Delay Time.

NOTE: The SNMP Trap actions are valid for Serial, Environmental, UPS and Cellular data triggers.

- Click Save New Action.

8.3.6 SEND NAGIOS EVENT

- Select Send Nagios Event as the Add Trigger Action.
- Enter a unique Action Name.
- Set the Action Delay Time.
- Edit the Nagios Event Message text to display on the Nagios status screen for the service.
- Specify the Nagios Event State (OK, Warning, Critical, or Unknown) to return to Nagios for this service.
- Click Save New Action.

NOTE: To notify the central Nagios server of Alerts, NSCA must be enabled under System > Nagios and Nagios must be enabled for each applicable host or port

8.3.7 PERFORM INTERFACE ACTION

- Select Perform Interface Action as the Add Trigger Action.
- Enter a unique Action Name.
- Set the Action Delay Time.
- Select the Interface (Modem or VPN Service).
- Select the Action (Start Interface or Stop Interface) to be taken.

  For example, you may wish to start an IPsec VPN service in response to an incoming SMS message, or set up an OpenVPN tunnel whenever your console server fails over to use the cellular connection.
- Click Save New Action.

NOTE: If any IPsec service or OpenVPN tunnel is to be controlled by the Network Interface Event Action, the Control by Auto-Response checkbox must be checked when configuring that service. Also, if selected, the default state for the VPN tunnel or service will be Down.

8.4 RESOLVE ACTIONS

Actions can also be scheduled to be taken when a trigger condition has been resolved.

- For a nominated Auto-Response with a defined trigger Check Condition, click Add Resolve Action (for example, Send Email or Run Custom Script) to select the action type to be taken.

NOTE: Resolve Actions are configured the same way as Trigger Actions except the designated Resolve Actions are all executed on resolution of the trigger condition and there are no Action Delay Times to set.

8.5 CONFIGURE SMTP, SMS, SNMP AND NAGIOS SERVICE FOR ALERT NOTIFICATIONS

The Auto-Response facility enables remote alerts to be sent as Trigger and Resolve Actions. Before such alert notifications can be sent, you must configure the nominated alert service.

8.5.1 SEND E-MAIL ALERTS

The console server uses SMTP (Simple Mail Transfer Protocol) for sending the email alert notifications. To use SMTP, the Administrator must configure a valid SMTP server for sending the email.

- Navigate to Alerts & Logging > SMTP & SMS > SMTP Server.
- Enter the IP address of the outgoing mail Server in the Server field.
- If this mail server uses a Secure Connection, select its type.
- Specify the IP port to use. The default SMTP Port is 25.
- Optionally enter a Sender email address. This will appear as the From address in all email notifications sent from this console server.

NOTE: Many SMTP servers check the sender's email address with the host domain name to verify the address as authentic. So it may be useful to assign an email address for the console server such as consoleserver2@mydomian.com.

- If the SMTP server requires authentication, enter the required Username and Password.
- Optionally, enter a Subject Line that will be sent with all email notifications.

NOTE: Some SMTP servers require a non-blank Subject field.

- Click Apply. SMTP is activated.

8.5.2 SEND SMS ALERTS

With any model console server, you can use email-to-SMS services to send SMS alert notifications to mobile devices. Almost all mobile phone carriers provide an SMS gateway service that forwards email to mobile phones on their networks. There's also a wide selection of SMS gateway aggregators that provide email to SMS forwarding to phones on any carriers. Alternately, if your console server has an embedded or externally attached cellular modem, you will be given the option to send the SMS directly over the carrier connection.

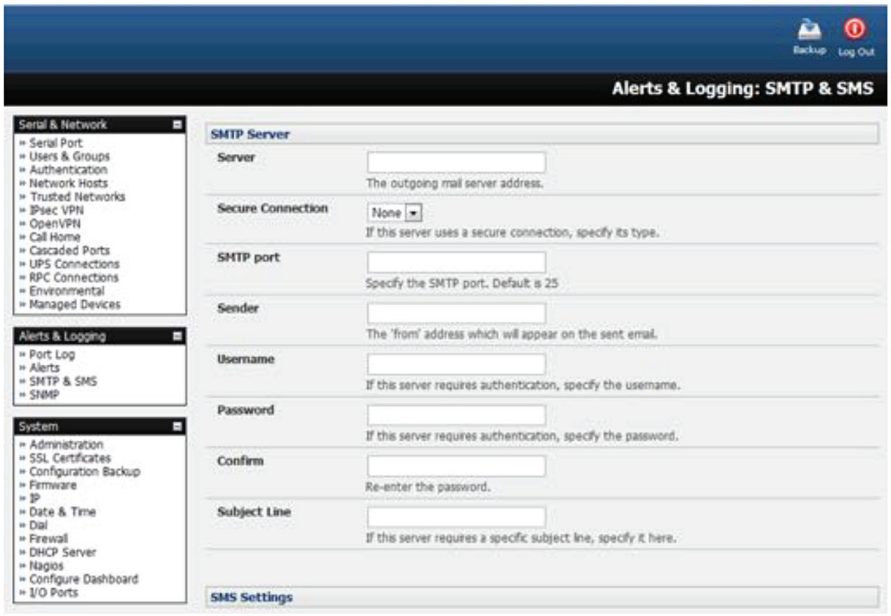


FIGURE 8-11. SMTP SERVER SCREEN

SMS via e-mail gateway

To use SMTP SMS, the Administrator must configure a valid SMTP server for sending the email.

- Navigate to Alerts & Logging > SMTP & SMS.
- Select the SMS Gateway radio button in the SMS Settings section. An SMS via Email Gateway section will appear.
- Enter the IP address of the outgoing SMS gateway Server.

FIGURE 8-12. SMS SETTINGS SCREEN

Select a Secure Connection (if applicable).

- Specify the SMTP port to be used. The default SMTP Port is 25.
- Optionally enter a Sender email address. This will appear as the From address in all email notifications sent from this console server.

NOTE: Some SMS gateway service providers only forward email to SMS when the email has been received from authorized senders. You may need to assign a specific authorized email address for the console server.

- If the SMTP server requires authentication, enter the required Username and Password.
- Optionally, enter a Subject Line that will be sent with all notifications.

NOTE: Generally, the email subject will contain a truncated version of the alert notification message (which is contained in full in the body of the email). Some SMS gateway service providers require blank subjects or require specific authentication headers to be included in the subject line.

- Click Apply Settings. The SMS-SMTP connection is activated.

SMS via cellular modem

To use an attached or internal cellular modem for SMS, the Administrator must enable SMS.

- Navigate to Alerts & Logging > SMTP & SMS.
- Select the Cellular Modem radio button in the SMS Settings section.
- Check Receive Messages to enable incoming SMS messages to be received.

A custom script will be called on receipt of incoming SMS messages.

You may need to enter the phone number of the carrier's SMS Message Center. Only enter this if advised by your carrier or by Black Box Technical Support.

- Click Apply Settings. The SMS-SMTP connection is activated.

NOTE: The option to directly send SMS alerts via the cellular modem was included in the Management GUI as of firmware v3.4. Advanced console servers have had the gateway software (SMS Server Tools 3) embedded since firmware v3.1, but you could only access this from the command line to send SMS messages.

FIGURE 8-13. SMS VIA CELLULAR MODEM SCREEN

8.5.3 SEND SNMP TRAP ALERTS

The Administrator can configure the Simple Network Management Protocol (SNMP) agent that resides on the console server to send SNMP trap alerts to an NMS management application.

- Navigate to Alerts & Logging > SNMP.
- Click the Primary SNMP Manager tab.

The Primary SNMP Manager and Secondary SNMP Manager tabs are used to configure where and how outgoing SNMP alerts and notifications are sent.

If you require your console server to send alerts via SNMP, a Primary SNMP Manager must be configured.

Optionally, a second SNMP Network Manager, with its own SNMP settings, can be specified on the Secondary SNMP Manager tab.

NOTE: Console servers can also be configured to provide status information on demand using snmpd. This SNMP agent is configured using the SNMP Service Detail at Alerts & Logging > SNMP. See Chapter 16 for more information.

- Select the Manager Protocol. SNMP is generally a UDP-based protocol though infrequently it uses TCP instead.
- Enter the host address of the SNMP Network Manager in the Manager Address field.
- Enter the TCP/IP port number into the Manager Trap Port field By default this port number is 162.
- Select the Version to be used. The console server SNMP agent supports SNMP v1, v2 and v3.
- •Enter the Community name for SNMP v1 or SNMP v2c. At a minimum, a community needs to be set for either SNMP v1 or v2c traps to work. An SNMP community is the group to which devices and management stations running SNMP belong. It helps define where information is sent. SNMP default communities are private for Write and public for Read.

- If required, configure SNMP v3. For SNMP v3 messages, the user's details and security level must match what the receiving SNMP Network Manager is expecting. SNMP v3 mandates that the message will be rejected unless the SNMPv3 user sending the trap already exists in the user database on the SNMP Manager. The user database in a SNMP v3 application is actually referenced by a combination of the Username and the Engine ID for the given SNMP application you are talking to.
- Enter the Engine ID for the user sending messages. This is a hex number. For example: 0x8000000001020304.
- Specify the Security Level. The security level has to be compatible with the settings of the remote SNMP Network Manager.

TABLE 8-2. SECURITY LEVEL OPTIONS

SECURITY LEVEL	DESCRIPTION
noAuthNoPriv	No authentication or encryption
authNoPriv	Authentication only. An authentication protocol (SHA or MD5) and password are required.
authPriv	Authentication and encryption. Requires an encryption protocol (DES or AES) and an authentication protocol password.

- Complete the Username. This is the Security Name of the SNMPv3 user sending the message. This field is mandatory and must be completed when configuring the console server for SNMPv3.
- If the required Security Level is authNoPriv or authPrive, select an Authentication Protocol (either SHA or MD5) and an Authentication Password. The password must contain at least 8 characters.
- If the required Security Level is authPriv, select a Privacy Protocol (DES or AES). AES is recommended. A password of at least 8 characters must be provided for encryption to work.
- Click Apply.

NOTE: Console servers with firmware v3.0 and later also embed the net-snmpd dæmon. This dæmon can accept SNMP requests from remote SNMP management servers and provides information on alert status, serial port status and device status (see Section 16.5 for more details). Console servers with firmware earlier than v3.3 can only configure a Primary SNMP server from the Management Console. See Section 16.5 for details on configuring the snmptrap daemon to send traps/notifications to multiple remote SNMP servers.

As of firmware v3.10.2, new SNMP status and trap MIBS were created to provide more and better structured SNMP status and traps from console servers.

There is an option in Alerts & Logging > SNMP to Use Legacy Notifications for the SNMP traps.

Setting this option sets the console server to SNMP traps that are compatible with those sent in older firmware before the new MIBS were added. Setting this option ensures a firmware upgrade to v3.10.2 or later does not break existing SNMP management.

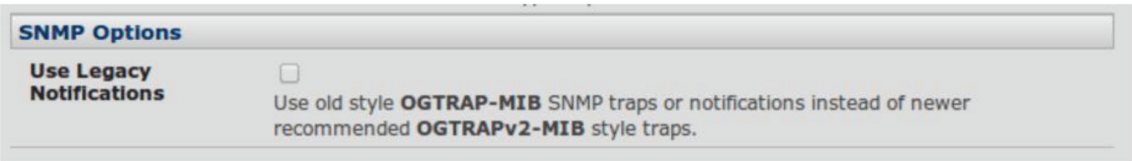


FIGURE 8-14. SNMP OPTION SCREEN

When upgrading from firmware that does not support the newer SNMP MIBs/traps (firmware versions before 3.10.2) to firmware that does support the new MIBs/traps:

If the SNMP service was enabled and an SNMP manager was configured before upgrading the firmware, the console server will be configured to use the legacy traps after upgrading.

If the SNMP service was not enabled or no SNMP manger was configured before the upgrade, the console server will be configured to use the new SNMP traps after the upgrade. This won't have any effect until the SNMP service is turned on and an SNMP manager is configured.

8.5.4 SEND NAGIOS EVENT ALERTS

To notify the central Nagios server of Alerts, NSCA must be enabled under System > Nagios and Nagios must be enabled for each applicable host or port under Serial & Network > Network Hosts or Serial & Network > Serial Ports (see Chapter 11).

NOTE: In a VCMS centrally managed environment you can check the Nagios alert option. On the trigger condition (for matched patterns, logins, power events and signal changes), an NSCA check warning result will be sent to the central Nagios server. This condition is displayed on the Nagios status screen and triggers a notification, which can then cause the Nagios central server itself to send out an email or an SMS, page, etc.

8.6 LOGGING

The console server can maintain log records of auto-response events. It can also log records of all access and communications events with both the console server and with attached serial, network and power devices.

A log of all system activity is also maintained by default, as is a history of the status of any attached environmental monitors.

8.6.1 LOG STORAGE

Before activating any Event, Serial, Network or UPS logging, you must specify where those logs are to be saved. These records are stored off-server or in the gateway USB flash memory.

- Navigate to Alerts & Logging > Port Log.

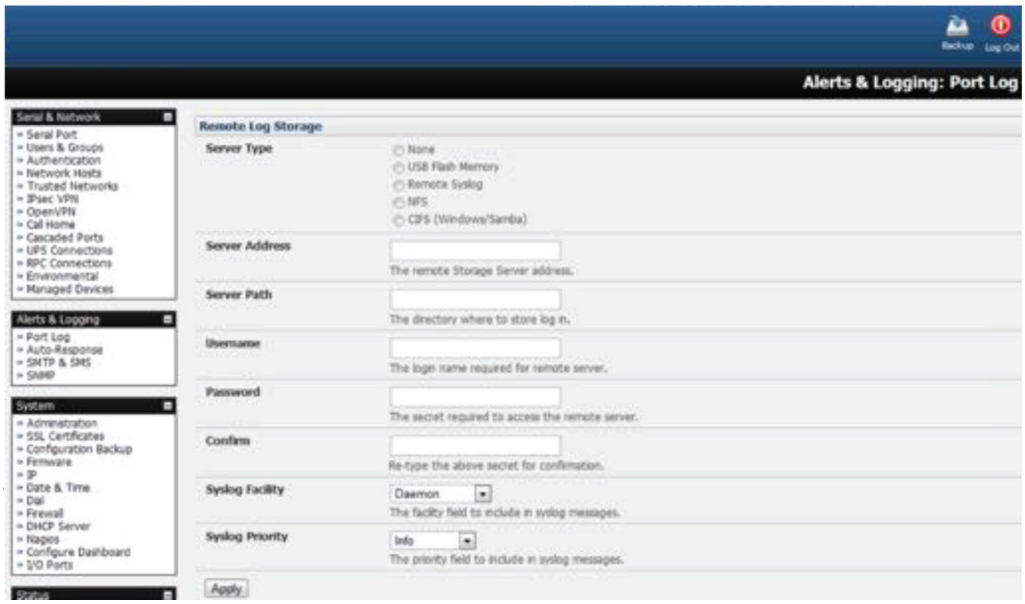


FIGURE 8-15. PORT LOG SCREEN

- Specify the Server Type to be used.
- Add the required server details to enable log server access.

The Administrator can view serial, network, and power device logs stored in the console reserve memory (or on a USB-connected flash device) in Manage > Devices.

A User will only see logs for the Managed Devices they (or their Group) have been given access privileges for (see Chapter 14).

View USB event logs in a web terminal or by ssh or telnet to the console server.

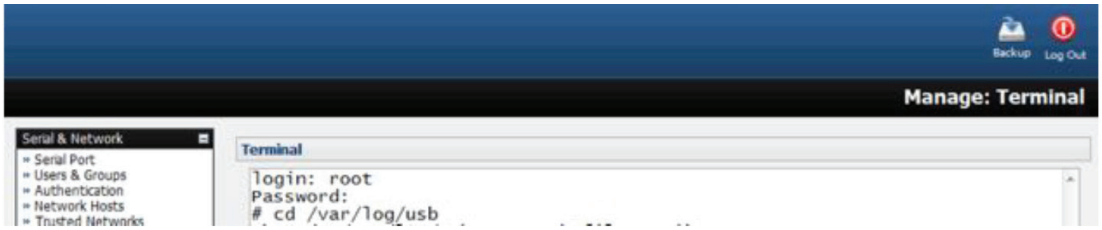


FIGURE 8-16. MANAGE TERMINAL SCREEN

8.6.2 SERIAL PORT LOGGING

In Console Server mode, activity logs can be maintained of all serial port activity. To specify which serial ports are to have activities recorded and to what level data is to be logged:

- Navigate to Serial & Network > Serial Port.
- Click Edit for the port to be logged.
- Specify the Logging Level for each port.

TABLE 8-3. LOGGING LEVEL OPTIONS

LEVEL	USER CONNECTON EVENTS	DATA TRANSFERRED TO THE PORT	DATA TRANSFERRED FROM THE PORT	HARDWARE FLOW CONTROL CHANGES
0	not logged	not logged	not logged	not logged
1	logged	not logged	not logged	not logged
2	logged	logged	logged	logged
3	logged	not logged	logged	logged
4	logged	logged	not logged	logged

NOTE: Logging levels are not a progression from no logging to all logging. Logging Level 0 is no logging, but Logging Level 4 is not "more" logging than Logging Level 3: these two levels, for example, are different but 4 is not a more comprehensive amount of logging than 3.

- Click Apply.

NOTE: In addition to the Logs which are transmitted for remote/USB flash storage, a cache of the most recent 8K of logged data per serial port is maintained locally. To view the local cache of logged serial port data select Manage > Port Logs.

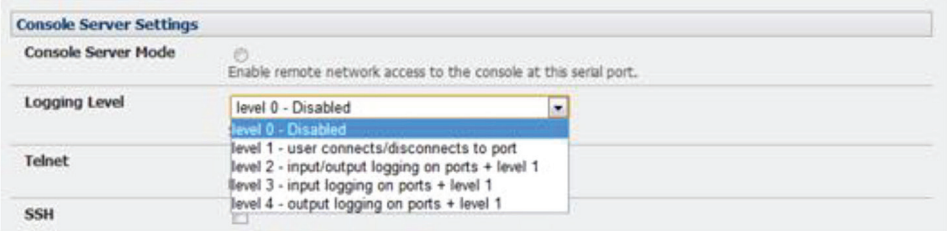


FIGURE 8-17. CONSOLE SERVER SETTINGS SCREEN

8.6.3 IP SUBNET-BASED VLAN

The console server supports optional logging of access to and communications with network attached Hosts.

For each Host, when you set up the Permitted Services that are authorized to be used, you also must set up the level of logging that is to be maintained for each service.

- Specify the logging level that is to be maintained for that particular TCP/UDP port/service, on that particular Host:

TABLE 8-4. LOGGING LEVEL OPTIONS

LOGGING LEVEL	DESCRIPTION
0	Turns off logging for the selected TCP/UDP port to the selected Host.
1	Logs all connection events to the port
2	Logs all data transferred to and from the port.

- Click Add.
- Click Apply.

8.6.4 AUTO-RESPONSE EVENT LOGGING

- Navigate to Alerts & Logging > Auto-Response.
- In the Global Auto-Response Settings section, check the Log Events check box.
- Click Save Settings.

8.6.5 POWER DEVICE LOGGING

The console server also logs access and communications with network attached hosts and maintain a history of the UPS and PDU power status.

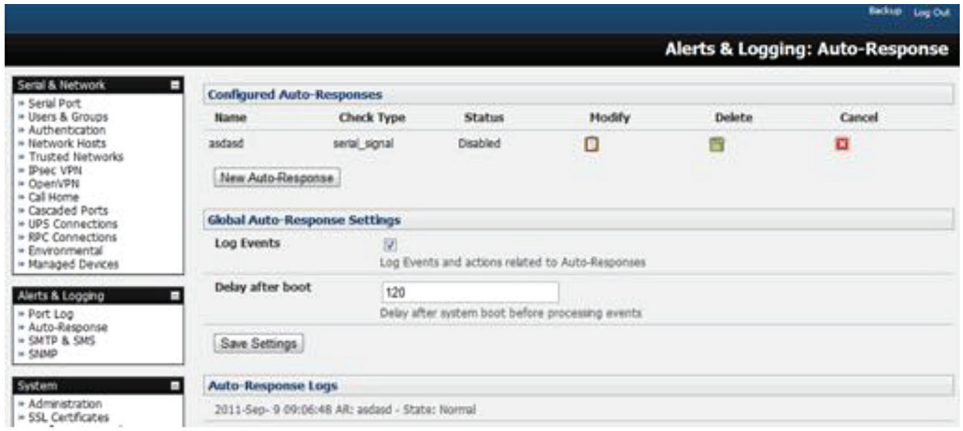


FIGURE 8-18.

To activate and set the desired levels of logging for UPS and PDU devices see Chapter 9.

CHAPTER 9: POWER, ENVIRONMENT AND DIGITAL I/O

Black Box console servers manage Remote Power Control devices (RPCs including PDUs and IPMI devices) and Uninterruptible Power Supplies (UPSes). They also monitor remote operating environments using Environmental Monitoring Devices (EMDs) and sensors, and can provide digital I/O control.

9.1 REMOTE POWER CONTROL (RPC)

The console server Management Console monitors and controls Remote Power Control (RPC) devices using the embedded PowerMan and Network UPS Tools open source management tools and Black Box's power management software. RPCs include power distribution units (PDUs) and IPMI power devices.

Serial PDUs invariably can be controlled using their command line console, so you could manage the PDU through the console server using a remote Telnet client. You could also use proprietary software tools supplied by the vendor. This generally runs on a remote Windows PC and you could configure the console server serial port to operate with a serial COM port redirector in the PC (as detailed in Chapter 5).

Similarly, network-attached PDUs can be controlled with a browser (with SDT as detailed in Section 7.3) or an SNMP management package or using the vendor supplied control software. Servers and network-attached appliances with embedded IPMI service processors or BMCs invariably are supplied with their own management tools (like SoL) that provide secure management when connected using SDT Connector.

For simplicity, all these devices can now all be controlled through the one window using the Management Console's RPC remote power control tools.

9.1.1 RPC CONNECTION

- Serial and network connected RPCs must first be connected to, and configured to communicate with, the console server.
- For serial RPCs, connect the PDU to the selected serial port on the console server.
- Navigate to Serial & Network > Serial Port.
- Configure the Common Settings of that port with the RS232 properties etc required by the PDU (see Section 5.1.1).
- Select RPC as the Device Type.
- Similarly for each network connected RPC, go to Serial & Network > Network Hosts and configure the RPC as a connected Host by specifying its Device Type as RPC.
- Click Apply.

FIGURE 9-1.

See Section 6.4 for more on Network Hosts.

- Navigate to Serial & Network > RPC Connections. The RPC connections that have already been configured will present.

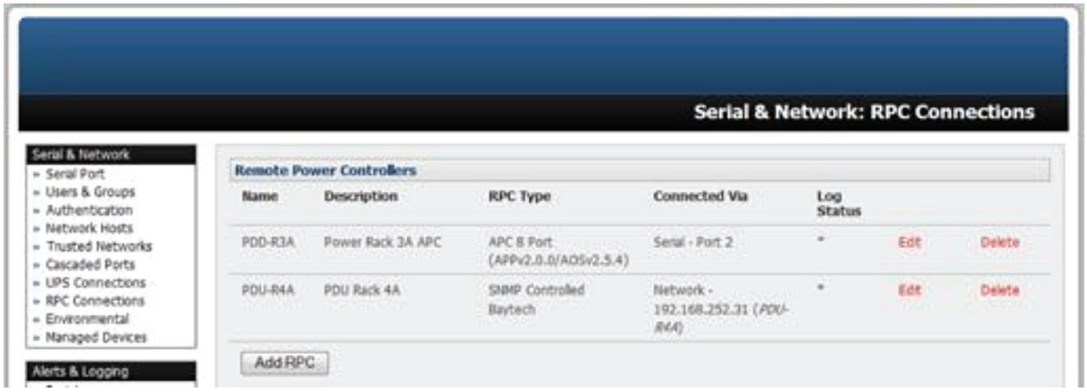


FIGURE 9-2.

- Click Add RPC.
- Connected Via presents a list of serial ports and network Host connections that you have set up with device type RPC but have yet to connect to a specific RPC device.
- When you select Connected Via for a Network RPC connection, the corresponding Host Name/Description set up for that connection will be entered as the Name and Description for the power device.

FIGURE 9-3.

- Alternatively, if you select Serial connection for Connected Via, you will need to enter a Name and Description for the power device.

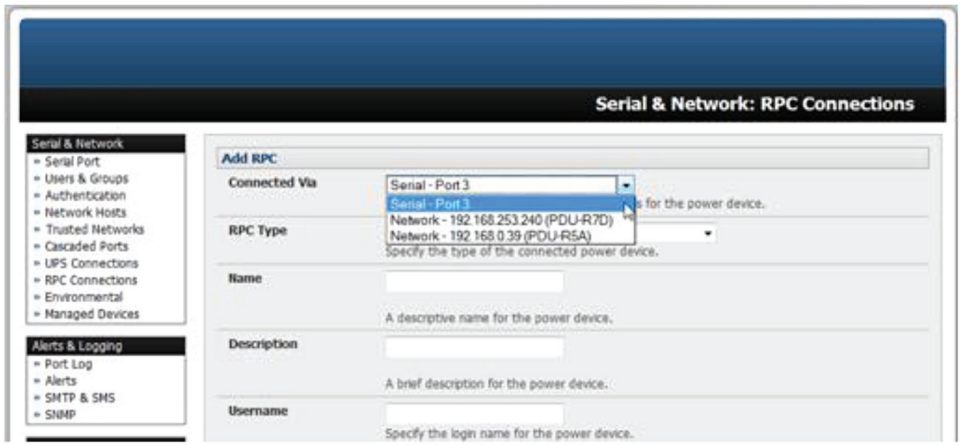


FIGURE 9-4.

- Select the appropriate RPC Type for the PDU (or IPMI) being connected.

  If you are connecting to the RPC via the network you will be presented with the IPMI protocol options and the SNMP RPC Types currently supported by the embedded Network UPS Tools.

  If you are connecting to the RPC by a serial port you will be presented with all the serial RPC types currently supported by the embedded PowerMan and Black Box's power manager.

- Enter the Username and Password used to login into the RPC.

NOTE: These login credentials are not related to the Users and access privileges configured in Serial & Networks > Users & Groups.

- If SNMP protocol is selected enter the SNMP v1 or v2c Community for Read/Write access. By default this would be private.
- Check Log Status.

FIGURE 9-5.

- Specify the Log Rate (minutes between samples) if you wish the status from this RPC to be logged. These logs can be views from Status > RPC Status.
- Click Apply.

For SNMP PDUs, the console server will now probe the configured RPC to confirm the RPC Type matches and will report the number of outlets it finds that can be controlled. If unsuccessful, it will report Unable to probe outlets and you will need to check the RPC settings, the network connection or the serial connection.

For serially connected RPC devices, a new Managed Device (with the same name as given to the RPC) will be created. The console server will then configure the RPC with the number of outlets specified in the selected RPC Type or will query the RPC itself for this information.

NOTE: Black Box's console servers support the majority of the popular network and serial PDUs. If your PDU is not on the default list then support can be added directly (see Chapter 16) or by having the PDU supported added to either the Network UPS Tools or PowerMan open source projects.

IPMI service processors and BMCs can be configured so all authorized users can use the Management Console to remotely cycle power and reboot computers, even when their operating system is unresponsive. To set up IPMI power control:

- Enter the IP address or domain name of the BMC or service processor (for example, a Dell DRAC) in Serial & Network > Network Hosts.
- Then, in Serial & Network > RPC Connections specify the RPC Type to be IPMI1.5 or 2.0.

9.1.2 RPC ACCESS PRIVILEGES AND ALERTS

Set PDU and IPMI alerts using Alerts & Logging > Alerts (see Chapter 8). Assign users to access and control outlets on each RPC via Serial & Network > User & Groups (see Chapter 5).

9.1.3 USER POWER MANAGEMENT

The Power Manager allows users and administrators to access and control configured serial- and network-attached PDU power strips, and servers with embedded IPMI processors or BMCs.

- Select Manage > Power.

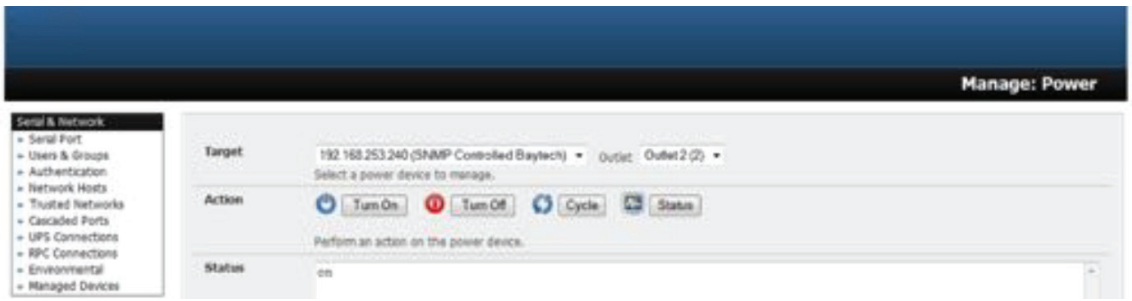


FIGURE 9-6.

- Select the Target power device to be controlled.
- If the RPC supports outlet level control, select the Outlet to be controlled.
- Initiate the desired Action by selecting the appropriate icon.

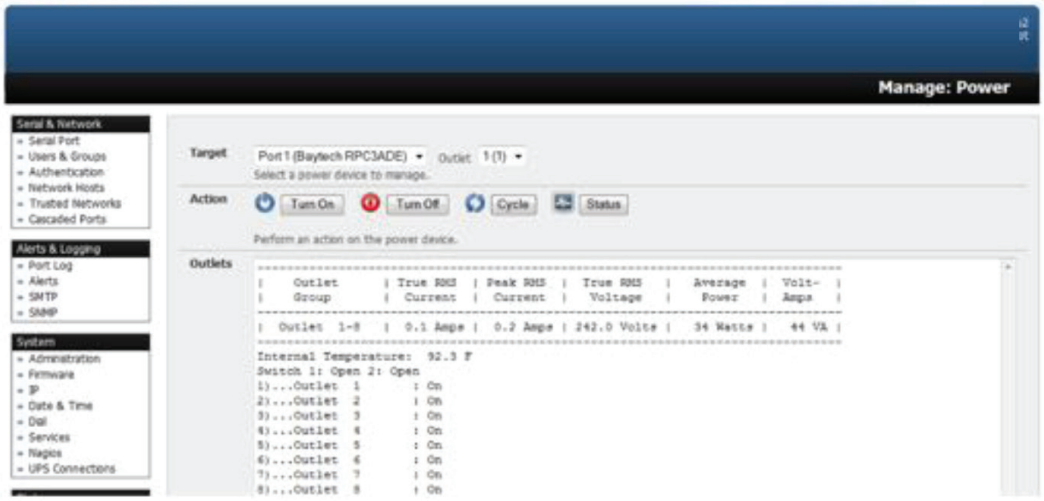


FIGURE 9-7.

NOTE: Icons will present only for operations that are supported by the Target you have selected.

Turn ON

Turn OFF

Cycle

Status

FIGURE 9-8.

9.1.4 RPC STATUS

You can monitor the current status of your network and serially connected PDUs and IPMI RPCs.

- Select Status > RPC Status. A table with the summary status of all connected RPC hardware will display.

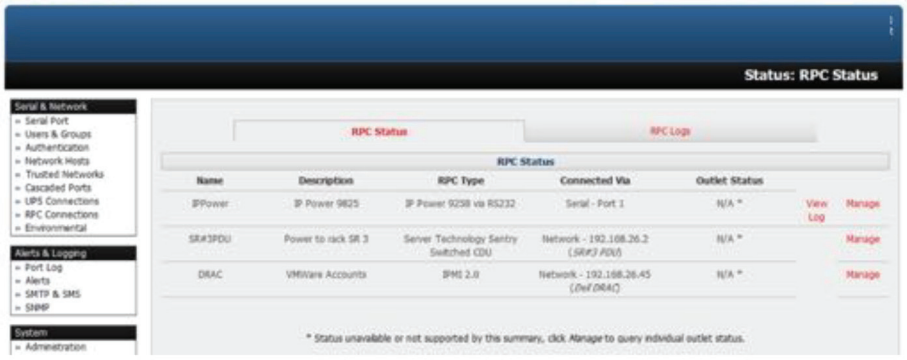


FIGURE 9-9.

- Click on View Log or select the RPCLogs tab. A table of the history and detailed graphical information on the selected RPC will present.

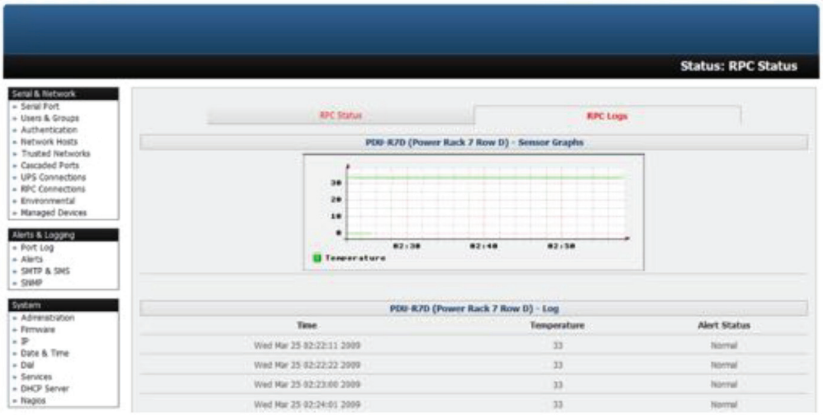


FIGURE 9-10.

- Click Manage to query or control the individual power outlet. This will take you to Manage > Power.

9.2 UNINTERRUPTIBLE POWER SUPPLY (UPS) CONTROL

Black Box console servers can be configured to manage locally and remotely connected UPS hardware using Network UPS Tools.

Network UPS Tools (NUT) is a group of open source programs that provide a common interface for monitoring and administering UPS hardware and ensuring safe shutdowns of the systems that are connected. NUT is built on a networked model with a layered scheme of drivers, server and clients. It is covered in some detail in Section 9.2.6.

9.2.1 MANAGED UPS CONNECTIONS

A Managed UPS is a UPS that is directly connected as a Managed Device to the console server. It can be connected by serial or USB cable or by the network. The console server becomes the master of this UPS, and runs a upsd server to allow other computers that are drawing power through the UPS (slaves) to monitor the UPS status and take appropriate action, such as shutdown, in event of low UPS battery.

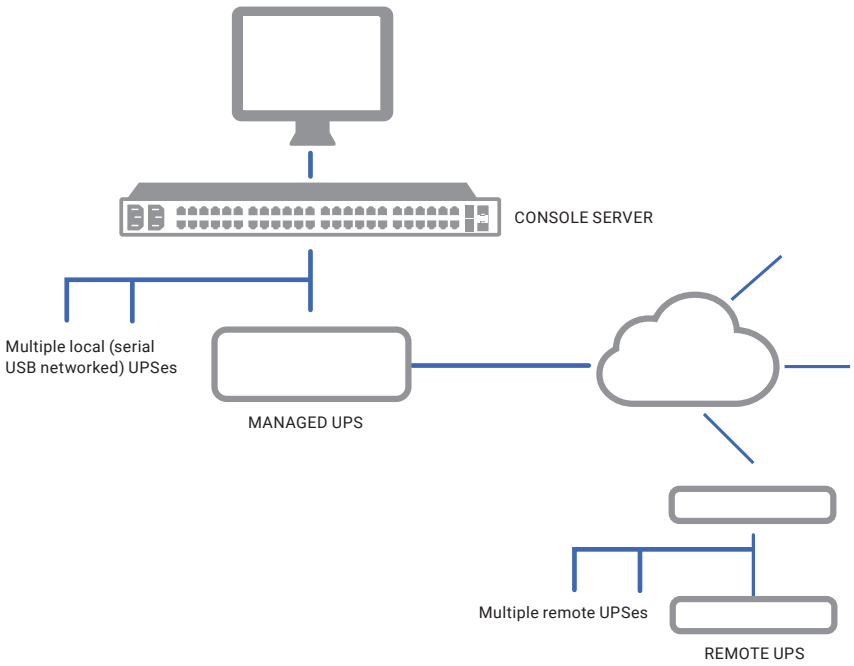


FIGURE 9-11.

The console server may or may not be drawing power itself through the Managed UPS. When the UPS's battery power reaches critical, the console server signals, waits for slaves to shut down, then powers off the UPS.

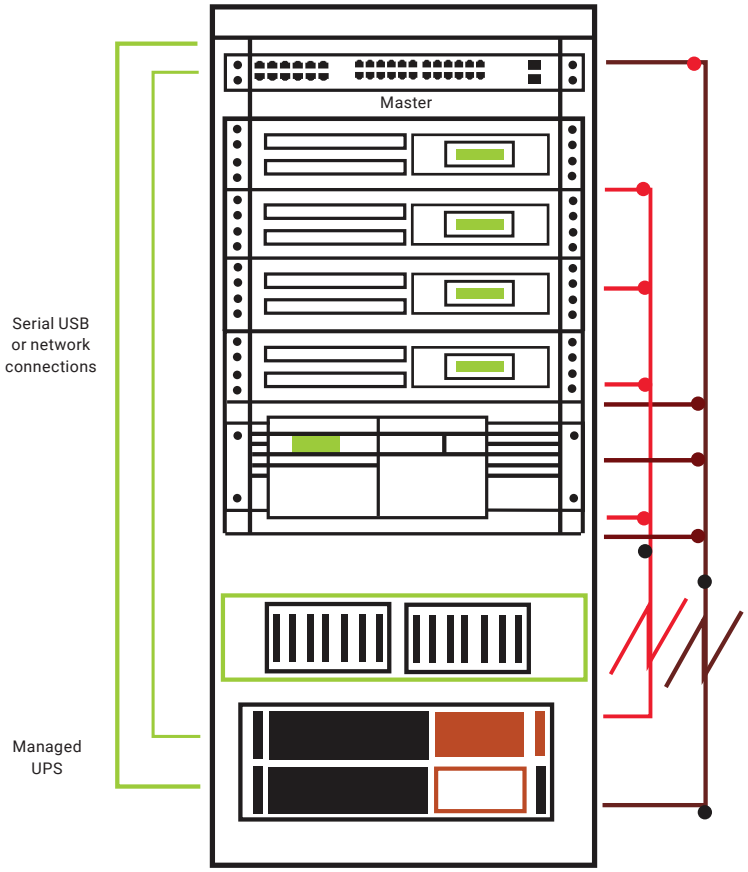


FIGURE 9-12.

Serial and network connected UPSes must first be connected to, and configured to communicate with the console server.

For serial UPSes attach the UPS to the selected serial port on the console server:

- Navigate to Serial and Network > Serial Port.
- Configure the Common Settings of that port with the properties (RS-232, etc.) required by the UPS (see Section 5.1.1).
- Select UPS as the Device Type.

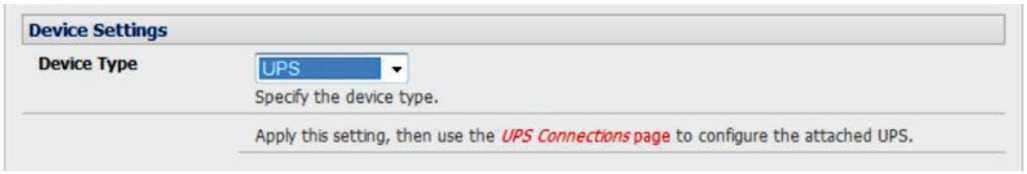


FIGURE 9-13.

For each network connected UPS:

- Navigate to Serial & Network > Network Hosts.
- Configure the UPS as a connected Host by specifying its Device Type as UPS.
- Click Apply.

NOTE: USB-connected UPS hardware requires no equivalent configuration.

FIGURE 9-14.

- Navigate to Serial & Network > UPS Connections. The Managed UPSes section will display all UPS connections which have already been configured.
- Click Add Managed UPS.
- Select if the UPS will be Connected Via USB or over a pre-configured serial port or via SNMP/HTTP/HTTPS over the preconfigured network Host connection.

NOTE: When you select a network UPS connection, the corresponding Host Name/Description that you set up for that connection will be entered as the Name and Description for the power device. Alternatively, if you select to Connect Via a USB or serial connection, you will need to enter a Name and Description for the power device. These details will also be used to create a new Managed Device entry for the serial/USB connected UPS devices.

- Enter the login details. This Username and Password is used by slaves of this UPS (other computers that are drawing power through this UPS) to connect to the console server to monitor the UPS status so they can shut themselves down when battery power is low.

  Monitoring will typically be performed using the upsmon client running on the slave server (see Section 9.2.3).

NOTE: These login credentials are not related to the Users and access privileges configured in Serial & Networks > Users & Groups.

- Select the action to take when UPS battery power becomes critical: Shut down the UPS or Shut down all Managed UPSes or simply Run until failure.

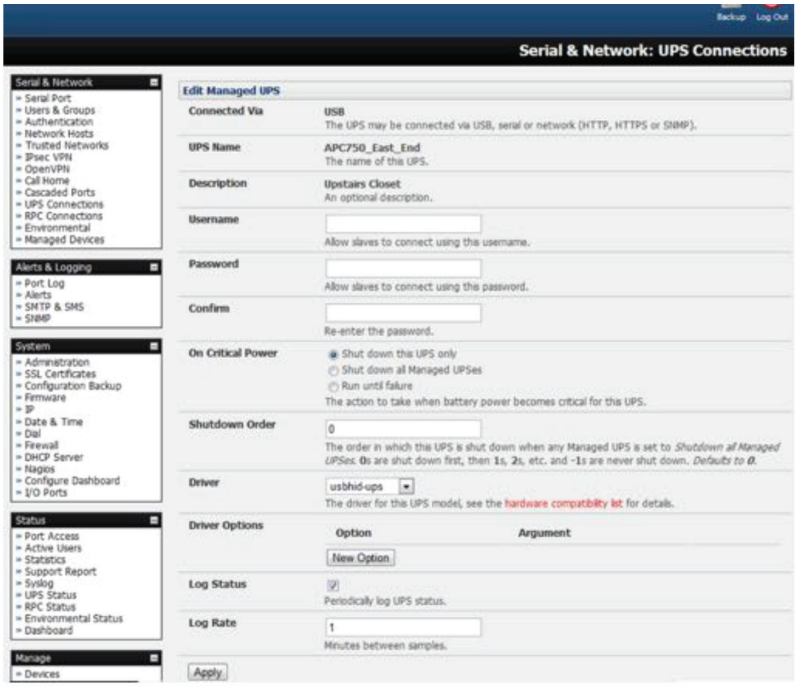


FIGURE 9-15.

The shutdown script /etc/scripts/ups-shutdown can be customized so, in the event of a critical power failure (when the UPS battery runs out) you can program the console server to perform last gasp actions before power is lost. It is generally simpler, however, to perform such last gasp actions by triggering Auto-Response on the UPS pressing batt or lowbatt. See Chapter 8.

- If you have multiple UPSes and require them to be shut down in a specific order, specify the Shutdown Order for this UPS.

  This is a whole positive number, a 0 or -1. 0s are shut down first, then 1s, 2s, 3s and so on. -1s are not shut down at all. The default value is 0.

- Select the Driver that will be used to communicate with the UPS

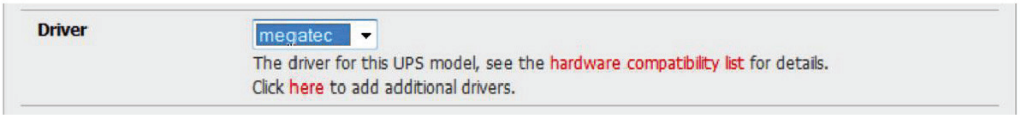


FIGURE 9-16.

- Click New Options in Driver Options if you need to set driver-specific options for your selected NUT driver and hardware combination.

FIGURE 9-17.

For more details see http://www.networkupstools.org/doc.

- Check Log Status and specify the Log Rate (minutes between samples) if you wish the status from this UPS to be logged.

  These logs can then be viewed at Status > UPS Status.

If you have enabled Nagios services you will presented with an option for Nagios monitoring

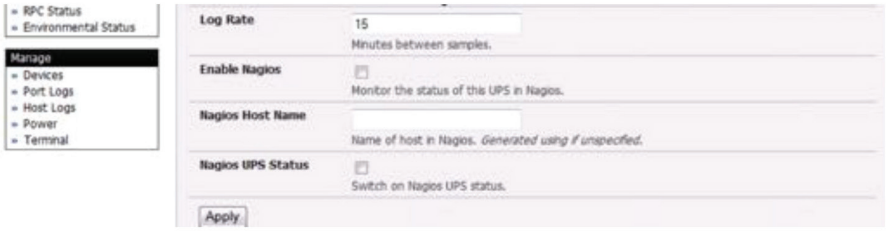


FIGURE 9-18.

- Check Enable Nagios to enable this UPS to be monitored using Nagios central management.
- Check Enable Shutdown Script if this is the UPS providing power to the console server itself.

  If there is a critical power failure, you can perform last gasp actions on the console server before power is lost.

  This is achieved by placing a custom script in /etc/config/scripts/ups-shutdown (you may use the provided /etc/scripts/ups-shutdown as a template). This script is only run when then UPS reaches critical battery status.
- Click Apply.

NOTE: You can customize the upsmon, upsd and upsc settings for this UPS hardware directly from the command line.

9.2.2 REMOTE UPS MANAGEMENT

A Remote UPS is a UPS that is connected as a Managed Device to some remote console server that is being monitored (but not managed) by your console server.

The upsc and upslog clients in the console server can configured to monitor remote servers that are running Network UPS Tools managing their locally connected UPSes. These remote servers might be other Black Box console servers or generic Linux servers running NUT. All these distributed UPSes (which may be spread in a row in a data center, or around a campus property or across the country) can be centrally monitored through the one central console server window.

An example where centrally monitoring remotely distributed UPSes is useful is a campus or large business complex where there's a multitude of computer and other equipment sites spread afar, each with their own UPS supply. Many of these (particularly the smaller sites) will be USB or serially connected.

Having a LES1200 or LES1508A at these remote sites allows the systems manager to centrally monitor the power supply status at all sites, centralize alarms, and, consequently, be warned to initiate a call-out or take shut down actions.

To add a Remote UPS:

- Navigate to Serial & Network > UPS. The Remote UPSes section displays all the remote UPS devices being monitored.

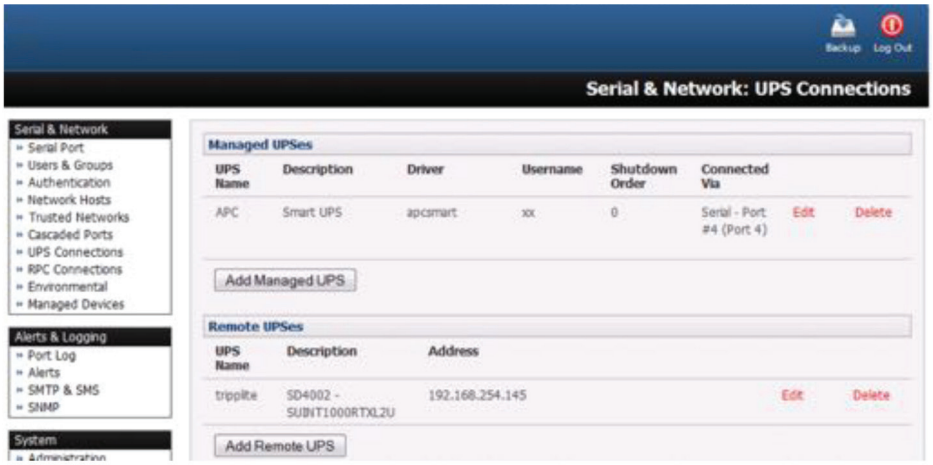


FIGURE 9-19.

- Click Add Remote UPS.

FIGURE 9-20.

- Enter the UPS Name of the remote UPS to be remotely monitored. This name must be the name that the remote UPS was configured with on the remote console server as the remote console server may itself have multiple UPSes attached that it is managing locally with NUT.
- Optionally enter a Description.
- Enter the IP Address or DNS name of the remote console server that is managing the remote UPS. This may be another Black Box console server or it may be a generic Linux server running Network UPS Tools.
- Check Log Status.
- Specify the Log Rate (minutes between samples) if you wish the status from this UPS to be logged. These logs can then be viewed at Status > UPS Status.

- Check Enable Shutdown Script if this remote UPS is the UPS providing power to the console server itself. If the UPS reaches critical battery status the custom script in /etc/config/scripts/ups-shutdown is run enabling you to perform any “last gasp” actions.
- Click Apply.

NOTE: The Remote UPS feature is supported on all console servers running firmware v2.8 and later. Earlier versions supported a single remote monitored UPS that could be set to trigger the console server shutdown script.

9.2.3 CONTROLLING UPS-POWERED COMPUTERS

One of the advantages of having a Managed UPS is that you can configure computers that draw power through that UPS to shut down gracefully in case of UPS problems.

For Linux computers, this can be done by setting up upsmon on each computer and directing them to monitor the console server that is managing their UPS.

This will set the specific conditions that will be used to initiate a power down of the computer. For example, non-critical servers may be powered down some seconds after the UPS starts running on battery where more critical servers may not be shut down until a low battery warning is received. Refer to the online NUT documentation for details on how this is done:

http://eu1.networkupstools.org/doc/2.2.0/INSTALL.html

http://linux.die.net/man/5/upsmon.conf

http://linux.die.net/man/8/upsmon

An example upsmon.conf entry might look like:

MONITOR managedups@192.168.0.1 1 username password slave

TABLE 9-1. UPSMON.CONF

UPSMON.CONF PORTION	DESCRIPTION
manageup	The UPS Name of the managed UPS.
192.168.0.1	The IP address of the Black Box console server.
1	Indicates the server has a single power supply attached to this UPS.
username	The username of the managed UPS.
password	The password of the managed UPS..

There are NUT monitoring clients available for Windows computers (for example, WinNUT).

If you have an RPC (PDU) it is also possible to shut down UPS-powered computers and other equipment without them have a client running (for example, communications and surveillance gear). Set up a UPS alert and use this to trigger a script that controls a PDU to shut off the power (see Chapter 16).

9.2.4 UPS ALERTS

Set UPS alerts using Alerts & Logging > Alerts. See Chapter 8.

9.2.5 UPS STATUS

You can monitor the current status of your network-connected, serially-connected or USB-connected Managed UPSes and any configured Remote UPSes.

- Navigate to Status > UPS Status. A table with the summary status of all connected UPS hardware will display.

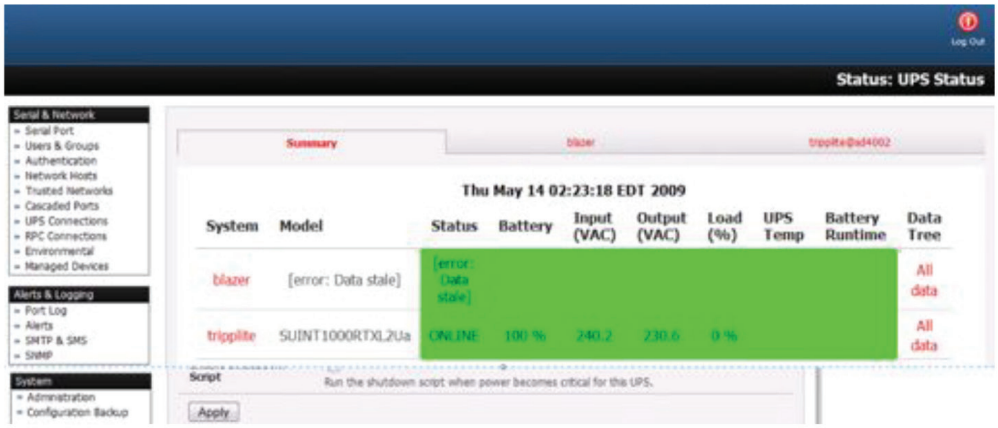


FIGURE 9-21.

- Click on any given UPS System name in the table. More detailed graphical information on the select UPS System will present.

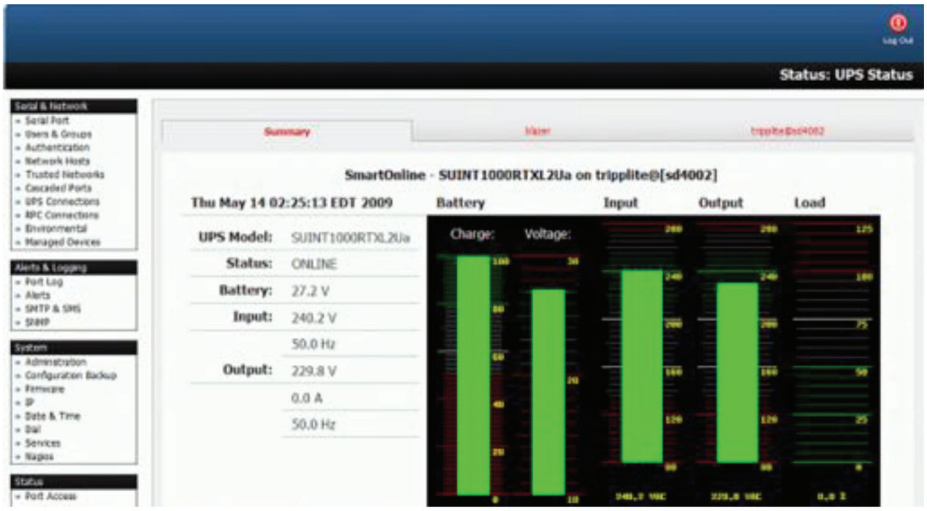


FIGURE 9-22.

- Click on any given UPS System's All Data link in the table. Status and configuration information on the selected UPS System presents.
- Select UPS Logs.

The log table of the load, battery charge level, temperature and other status information from all the managed and monitored UPS systems is presented.

This information is logged for all UPSes that were configured with Log Status checked. The information is also presented graphically.

9.2.6 OVERVIEW OF NETWORK UPS TOOLS (NUT)

Network UPS Tools (NUT) is built on a networked model with a layered scheme of drivers, server and clients. NUT can be configured using the Management Console as described above, or you can configure the tools and manage the UPSes directly from the command line. This section provides an overview of NUT.

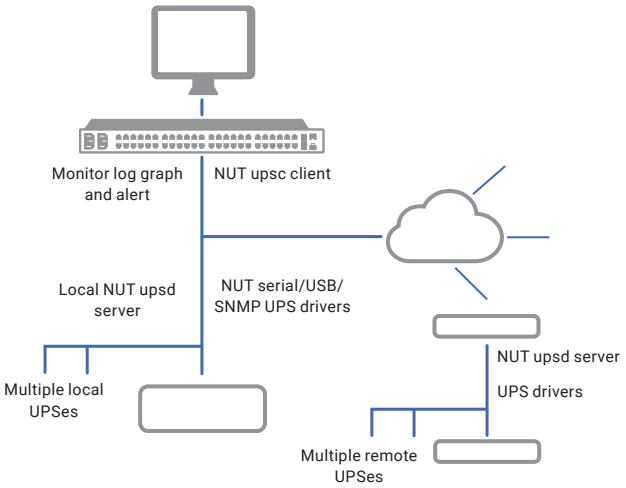


FIGURE 9-23.

The driver programs talk directly to the UPS equipment and run on the same host as the NUT network server (upsd). Drivers are provided for a wide assortment of equipment from most of the popular UPS vendors and understand the specific language of each UPS. They communicate to serial-, USB- and SNMP network- connected UPS hardware and map the communications back to a compatibility layer. This means both an expensive "smart" protocol UPS and a simple "power strip" model can be handled transparently.

The NUT network server program upsd is responsible for passing status data from the drivers to the client programs via the network. upsd can cache the status from multiple UPSes and then serve this status data to many clients. upsd also contains access control features to limit the abilities of the clients (for example, so only authorized hosts may monitor or control the UPS hardware).

There are a number of NUT clients that connect to upsd to check on the status of the UPS hardware and do things based on the status. These clients can run on the same host as the NUT server or they can communicate with the NUT server over the network (enabling them to monitor any UPS anywhere).

- The upsc client provides a quick way to poll the status of a UPS server. It can be used inside shell scripts and other programs that need UPS data but don't want to include the full interface.
- The upsmon client enables servers that draw power through the UPS to shutdown gracefully when the battery power reaches critical.
- There are also logging clients (upslog) and third party interface clients (Big Sister, Cacti, Nagios, Windows and more).

The latest release of NUT (2.7.4) also controls PDU systems. It can do this either natively using SNMP or through a binding to Powerman (open source software from Livermore Labs that is also embedded in Black Box console servers).

These NUT clients and servers are all embedded in each Black Box console server (with a Management Console presentation layer added). They also run remotely on distributed console servers and other remote NUT monitoring systems. This layered distributed NUT architecture enables:

- Multiple manufacturer support. NUT can monitor UPS models from 79 different manufacturers and PDUs from a growing number of vendors, all via a unified interface.
- Multiple architecture support. NUT can manage serial- and USB-connected UPS models with the same common interface. Network-connected USB and PDU equipment can also be monitored using SNMP.

- Multiple clients monitoring the one UPS. Multiple systems may monitor a single UPS using only their network connections. As well there is a wide selection of client programs that support monitoring UPS hardware via NUT (Big Sister, Cacti, Nagios and more).
- Central management of multiple NUT servers. A central NUT client can monitor multiple NUT servers that may be distributed throughout the data center, across a campus or around the world.

NUT supports the more complex power architectures found in data centers, communications centers and distributed office environments where UPSes from many vendors power many systems with many clients and larger UPSes power multiple devices and many of these UPSes are, in turn, dual powered.

9.3 DIGITAL I/O PORTS

LES1200 -I models and LES1508A -I models have four digital interface ports which present on a green connector block on the side of the unit.

DIO1 and DIO2 are two TTL level digital I/O ports: 5V max @ 20mA.

OUT1 and OUT2 are two High-Voltage digital output ports: >5V to <= 30V @100mA.

LES1600 models ship with a built-in, black, spring cage I/O connector block for attaching environmental sensors and digital I/O devices.

These I/O ports are configured via System > I/O Ports. Each port can be configured with a default direction and state.

- Navigate to System > I/O Ports.

9.3.1 DIGITAL I/O OUTPUT CONFIGURATION

Each of the two digital I/O ports (DIO1 and DIO2) can be configured as an Input or Output port. To use them as digital outputs, first configure the port direction on System > I/O Ports.

The DIO1 and DIO2 pins are current limited by the chip to 20 mA and accept 5 V levels, so they cannot, for example, drive a relay.

You can change the output states using the ioc command line utility. The following text is the ioc help text (also available by running ioc --help):

```
-p      pin_num  pin number (1 to 4)
-d      pin_dir  pin direction (0 = output 1 = input)
-v      pin_val  pin electrical value in output mode \  (0 = low 1 = high)
-r      reset pins to all inputs and low
-g      displays the pin directions and current values
-l      load pin configuration
```

For example, to set pin 1 to a low output, type:

```
ioc -p 1 -d 0 -v 0
```

To pulse one of these outputs, use a script like the following:

```
ioc -p 1 -d 0 -v 1
sleep 1
ioc -p 1 -d 0 -v 0
```

This sets the output high for 1 second, then returns it to low (assuming the initial state is low).

9.3.2 DIGITAL I/O INPUT CONFIGURATION

When either of the two digital I/O (DIO1 & DIO2) outlets is configured as an Input on the System > I/O Ports, it can be used to monitor the current status of any attached sensor.

When configured as inputs (the factory default), these first two ports are attached to an internal EMD. To configure them as alarms, go to the Status > Environmental Status and edit and enable the Internal EMD.

NOTE: The low-voltage circuits in DIO1 and DIO2 should not be wired to voltages greater than 5 VDC.

Alternatively, these input ports can be monitored using the ioc command line utility (as detailed in Section 9.4.1).

9.3.3 HIGH-VOLTAGE OUTPUTS

OUT1 and OUT2 (internally, DIO3 & DIO4) outlets are wired as high-voltage outputs. The way these outputs are expected to be used is to pull a power connected line to ground (the OUT1 and OUT2 transistors are open collector).

The I/O port header includes a 12-V reference line (VIN) which can be used to detect the line state change.

For example, to light a 12-V LED using the high voltage outputs, connect the positive leg of the LED to the 12-V reference, and the negative leg to output pin 4. Due to the way that the I/O port is connected internally, the output has to be set high to pull the output to ground.

The following command will switch on the led:

```
ioc -p 4 -d 0 -v 1
```

OUT1 and OUT2 transistors can operate with a supply of >5V to <= 30V @100mA. This means to drive a relay circuit you must guarantee it doesn't provide more than 100mA when set to 1.

9.3.4 HIGH-VOLTAGE OUTPUTS

As of firmware v3.9, there is a SNMP status table that reports on the status of the digital IO ports.

The table's OID is OG-STATUSv2-MIB::ogEmdDioTable. Performing an snmpwalk on this table on a console server with DIO produces something like the following (the specifics will vary depending on device status):

```
$ snmpwalk -v2c -c public -M $MIBSDIR -m ALL t5:161 1.3.6.1.4.1.25049.16.5
OG-STATUS-MIB::ogDioStatusName.1 = STRING: DIO 1
OG-STATUS-MIB::ogDioStatusName.2 = STRING: DIO 2
OG-STATUS-MIB::ogDioStatusName.3 = STRING: DIO 3
OG-STATUS-MIB::ogDioStatusName.4 = STRING: DIO 4
OG-STATUS-MIB::ogDioStatusType.1 = INTEGER: ttlInputOutput(0)
OG-STATUS-MIB::ogDioStatusType.2 = INTEGER: ttlInputOutput(0)
OG-STATUS-MIB::ogDioStatusType.3 = INTEGER: highVoltageOutput(1)
OG-STATUS-MIB::ogDioStatusType.4 = INTEGER: highVoltageOutput(1)
OG-STATUS-MIB::ogDioStatusDirection.1 = INTEGER: input(1)
OG-STATUS-MIB::ogDioStatusDirection.2 = INTEGER: input(1)
OG-STATUS-MIB::ogDioStatusDirection.3 = INTEGER: input(1)
OG-STATUS-MIB::ogDioStatusDirection.4 = INTEGER: input(1)
OG-STATUS-MIB::ogDioStatusState.1 = INTEGER: low(0)
OG-STATUS-MIB::ogDioStatusState.2 = INTEGER: high(1)
```

OG-STATUS-MIB::ogDioStatusState.3 = INTEGER: high(1)

OG-STATUS-MIB::ogDioStatusState.4 = INTEGER: high(1)

OG-STATUS-MIB::ogDioStatusCounter.1 = Counter64: 0

OG-STATUS-MIB::ogDioStatusCounter.2 = Counter64: 0

OG-STATUS-MIB::ogDioStatusCounter.3 = Counter64: 0

OG-STATUS-MIB::ogDioStatusCounter.4 = Counter64: 0

OG-STATUS-MIB::ogDioStatusTriggerMode.1 = INTEGER: risingFallingEdge(3)

OG-STATUS-MIB::ogDioStatusTriggerMode.2 = INTEGER: risingFallingEdge(3)

OG-STATUS-MIB::ogDioStatusTriggerMode.3 = INTEGER: risingFallingEdge(3)

OG-STATUS-MIB::ogDioStatusTriggerMode.4 = INTEGER: risingFallingEdge(3)

CHAPTER 10: AUTHENTICATION

The console server platform is a dedicated Linux computer, and it embodies a myriad of popular and proven Linux software modules for networking, secure access (OpenSSH), secure communications (OpenSSL) and sophisticated user authentication (PAM, RADIUS, TACACS+, Kerberos and LDAP).

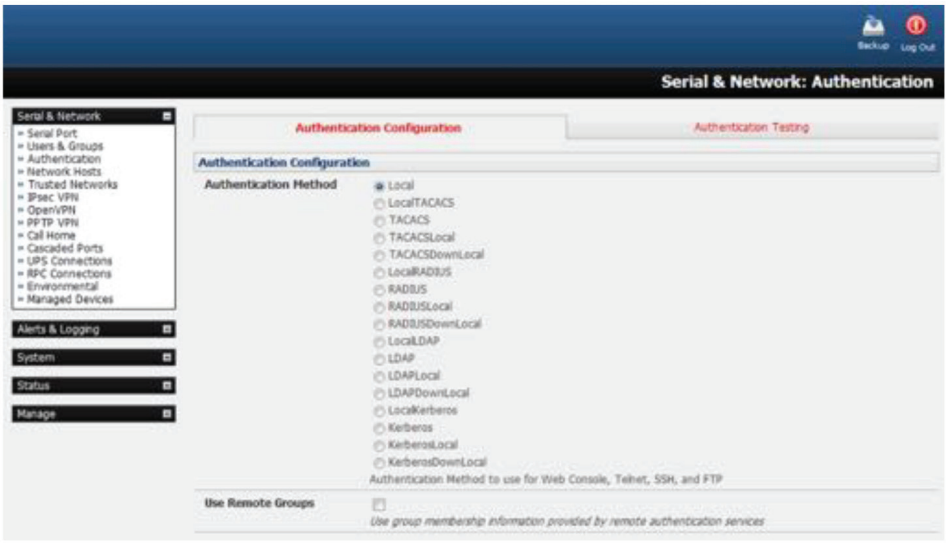


FIGURE 10-1.

This chapter details how the Administrator can use the Management Console to establish remote AAA authentication for all connections to the console server and attached serial and network host devices.

This chapter also covers establishing a secure link to the Management Console using HTTPS and using OpenSSL and OpenSSH for establishing secure Administration connection to the console server.

10.1 AUTHENTICATION CONFIGURATION

Authentication can be performed locally, or remotely using an LDAP, Radius, Kerberos or TACACS+ authentication server. The default authentication method for the console server is Local.

Any authentication method that is configured will be used for authentication of any user who attempts to log in through Telnet, SSH or the Web Manager to the console server and any connected serial port or network host devices.

The console server can be configured to the default (Local) or an alternate authentication method (TACACS, RADIUS, LDAP or Kerberos) with the option of a selected order in which local and remote authentication is to be used.

- Local/TACACS/RADIUS/LDAP/Kerberos

  Tries local authentication first, falling back to remote if local fails.

- TACACS/RADIUS/LDAP/Kerberos Local

  Tries remote authentication first, falling back to local if remote fails.

- TACACS/RADIUS/LDAP/Kerberos Down/Local

  Tries remote authentication first, falling back to local if the remote authentication returns an error condition (e.g., the remote authentication server is down or inaccessible).

10.1.1 LOCAL AUTHENTICATION

- Navigate to Serial and Network > Authentication.
- Check Local.
- Click Apply.

10.1.2 TACACS AUTHENTICATION

Perform the following procedure to configure the TACACS+ authentication method to be used whenever the console server or any of its serial ports or hosts is accessed.

- Select Serial and Network > Authentication.
- Check TACACS, LocalTACACS, TACACSLocal or TACACSDownLocal.
- Enter the Server Address (IP or host name) of the remote Authentication/Authorization server. Multiple remote servers may be specified in a comma separated list. Each server is tried in succession.
- Session accounting is on by default. If session accounting information is not wanted, check the Disable Accounting checkbox.

  One reason for not wanting session accounting: if the authentication server does not respond to accounting requests, said request may introduce a delay when logging in.

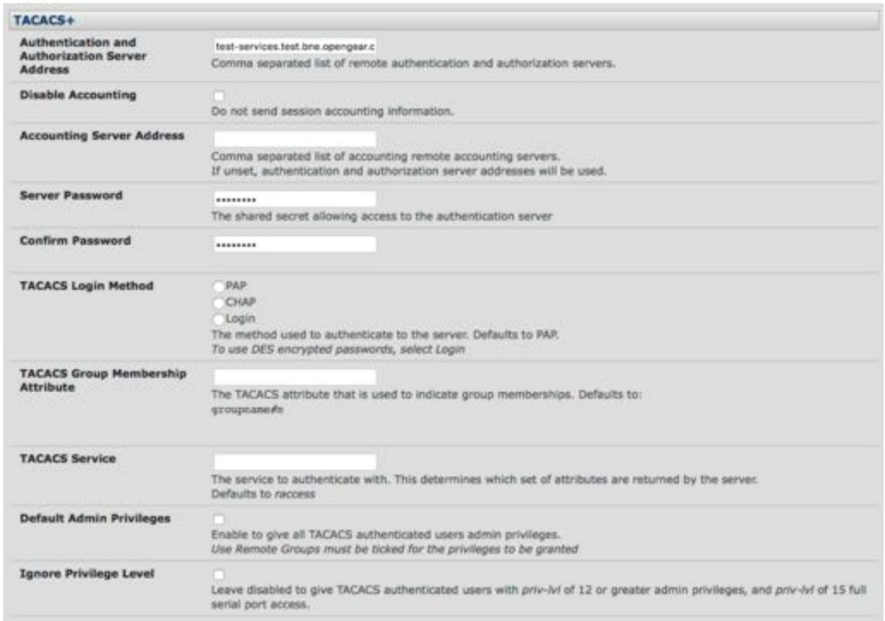


FIGURE 10-2.

- In addition to multiple remote servers, you can also enter separate lists of Authentication/Authorization servers and Accounting servers.

  If no Accounting servers are specified, the Authentication/Authorization servers are used instead.
- Enter and confirm the Server Password.
- Select the method to be used to authenticate to the server (defaults to PAP).

  To use DES encrypted passwords, select Login.

- If required, enter the TACACS Group Membership Attribute to be used to indicate group memberships (defaults to groupname#n).
- If required, specify TACACS Service to authenticate with.

  This determines which set of attributes are returned by the server (defaults to raccess).
- If required, check Default Admin Privileges to give all TACACS+ authenticated users admin privileges.
- Use Remote Groups must also be ticked for these privileges to be granted.
- The TACACS Privilege Level feature only applies to TACACS remote authentication.

  When Ignore Privilege Level is enabled, the priv-lvl setting for all of the users defined on the TACACS AAA server will be ignored.

NOTE: A console server normally interprets a user with a TACACS priv-lvl of 12 or above as an admin user. There is a special case where a user with a priv-lvl of 15 is also given access to all configured serial ports. When the Ignore Privilege Level option is enabled (that is, it is checked in the UI) there are no escalations of privileges based on the priv-lvl value from the TACACS server. If the only thing configured for one or more TACACS users is the priv-lvl (for example, no specific port access or group memberships are set), then enabling this feature will revoke access to the console server for those users as the they won't be a member of any groups, even if the Retrieve Remote groups option in the Authentication menu is enabled.

- Click Apply.

TACACS+ remote authentication will now be used for all user access to console server and serially or network attached devices.

The Terminal Access Controller Access Control System (TACACS+) security protocol is a protocol developed by Cisco. It provides detailed accounting information and flexible administrative control over the authentication and authorization processes.

TACACS+ allows for a single access control server (the TACACS+ daemon) to provide authentication, authorization, and accounting services independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

10.1.3 RADIUS AUTHENTICATION

The Remote Authentication Dial-In User Service (RADIUS) protocol was developed by Livingston Enterprises as an access server authentication and accounting protocol.

The RADIUS server can support a variety of methods to authenticate a user.

When it is provided with the username and original password given by the user, it can support PPP, PAP or CHAP, UNIX login, and other authentication mechanisms.

More information on configuring remote RADIUS servers can be found https://freeradius.org/ and https://cisco.com/c/en/us/support/docs/security-vpn/remote-authentication-dial-user-service-radius/12433-32.html.

Perform the following procedure to configure the RADIUS authentication method to be used whenever the console server or any of its serial ports or hosts is accessed:

- Select Serial & Network > Authentication and check RADIUS, LocalRADIUS, RADIUSLocal or RADIUSDownLocal.
- Enter the Server Address (IP or host name) of the remote Authentication and Authorization server. Multiple remote servers may be specified in a comma separated list. Each server is tried in succession.
- Session accounting is on by default. If session accounting information is not wanted, check the Disable Accounting checkbox.

  One reason for not wanting session accounting: if the authentication server does not respond to accounting requests, said request may introduce a delay when logging in.

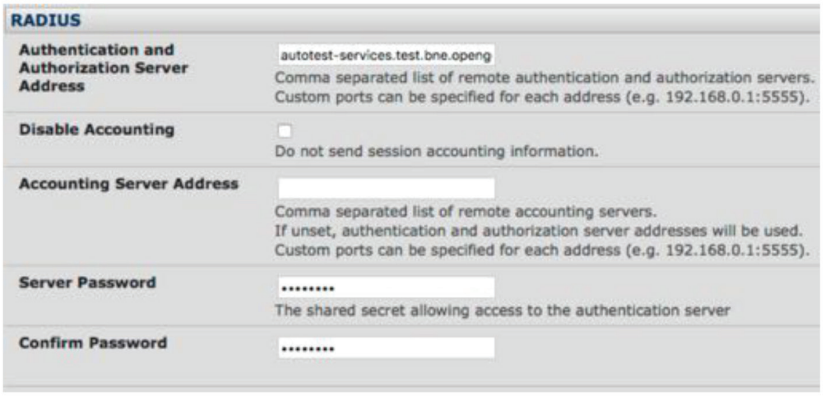


FIGURE 10-3.

In addition to multiple remote servers, you can also enter separate lists of Authentication and Authorization servers and Accounting servers.

If no Accounting servers are specified, the Authentication and Authorization servers are used instead.

- Enter the Server Password.
- Click Apply.

RADIUS remote authentication will now be used for all user access to console server and serially or network attached devices.

10.1.4 LDAP AUTHENTICATION

The Lightweight Directory Access Protocol (LDAP) is based on the X.500 standard, but is significantly simpler and more readily adapted to meet custom needs. The core LDAP specifications are all defined in RFCs. LDAP is a protocol used to access information stored in an LDAP server.

With firmware v3.11 and later, LDAP authentication now supports OpenLDAP servers, using the POSIX -style schema for user and group definitions.

Performing simple authentication against any LDAP server (AD or OpenLDAP) is straight forward, as they both follow the common LDAP standards and protocols. The harder part is configuring how to get the extra data about the users (for example, the groups they are in).

On an Black Box device, we may be configured to look at group information from an LDAP server for authentication and authorization. This group information is potentially stored in a number of different ways. Active Directory has one method; OpenLDAP has two others.

Active Directory method

Each entry for a user will have multiple memberOf attributes. Each memberOf value is the full DN of the group they belong to. (The entry for the user will be of objectClass user.)

OpenLDAP/POSIX method 1

Each entry for a user must have a gidNumber attribute. This will be an integer value, which is the user's primary group (for example, mapping to the /etc/passwd file, with the group ID field).

To determine which group this is, search for an entry in the directory that has that group ID, which will give the group name. (The users are of objectClass posixAccount, and the groups are of objectClass posixGroup.)

OpenLDAP/POSIX method 2

Each group entry in the group tree of objectClass posixGroup may have multiple memberUid attributes. These represent secondary groups (for example, mapping to the /etc/groups file). Each attribute would contain a username.

To cater for all these possibilities, the pam_ldap module has been modified to do group lookups for each of these three styles. This allows us to have a relatively generic configuration, and not be concerned with how the LDAP directory is set up.

There are only two parameters that need to be configured, based on what the user wishes to look up: the LDAP username and group membership attributes.

To clarify to the user what parameters to use, the descriptions for these fields have been updated to prompt the user for common or likely attributes. For example, the two configuration fields have descriptions as follows:

LDAP Username Attribute: the LDAP attribute that corresponds to the login name of the user (commonly 'sAMAccountName' for Active Directory, and 'uid' for OpenLDAP).

LDAP Group Membership Attribute: the LDAP attribute that indicates group membership in a user record (commonly 'memberOf' for Active Directory, and unused for OpenLDAP).

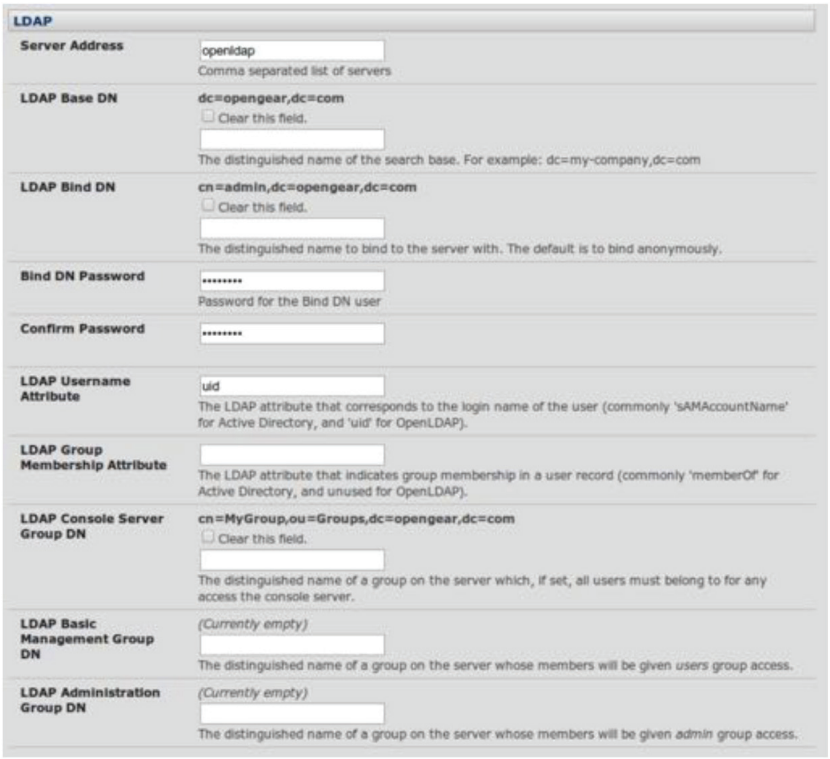


FIGURE 10-4.

NOTE: The libldap library is fussy about ensuring SSL connections are using certificates signed by a trusted CA. Consequently it is often not easy to set up a connection to an LDAP server using SSL.

Perform the following procedure to configure the LDAP authentication method to be used whenever the console server or any of its serial ports or hosts is accessed:

- Navigate to Serial & Network > Authentication.
- Check LDAP or LocalLDAP or LDAPLocal or LDAPDownLocal.

- Enter the Server Address (IP or host name) of the remote Authentication server. Multiple remote servers may be specified in a comma separated list. Each server is tried in succession.
- Check the Server Protocol checkbox to select if SSL is to be used or enforced for communications with the LDAP server.

  Console servers running firmware v3.11 and above offer three options for LDAPS (LDAP over SSL):

  LDAP over SSL preferred will attempt to use SSL for authentication. If it fails, it will fall back to LDAP without SSL.

  LDAP over SSL may fail due to certificate errors or the LDAP server not being contactable on the LDAPS port.

  LDAP over SSL only. This setting will configure the console server to only accept LDAP over SSL. If LDAP over SSL fails, you will only be able to log into the console server as root.

  LDAP (no SSL) only. This setting will configure the console server to only accept LDAP without SSL. If LDAP without SSL fails, you will only be able to log into the console server as root.
- Check the Ignore SSL Certificate Error check box if you wish to ignore SSL certificate errors, allowing LDAP over SSL to work regardless of these errors.

  This allows you to use any certificate, self-signed or otherwise, on the LDAP server without having to install any certificates on the console server.

  If this setting is not checked, you must install the CA (certificate authority) certificate with which the LDAP server's certificate was signed onto the console server. For example, the LDAP server is serving with a certificate signed using the certificate myCA.crt.

NOTE: The certificate must be in CRT format and myCA.crt must be installed onto the console server at /etc/config/ldaps_ca.crt. The filename must be ldaps_ca.crt. Copy the file to this location and filename manually using scp or the like. For example:

scp /local/path/to/myCA.crt

rt root@console_server:/etc/config/ldaps_ca.crt

- Enter the Server Password.
- Click Apply.

LDAP remote authentication will now be used for all user access to console server and serially or network attached devices

Further information on configuring remote RADIUS servers can be found at the following sites: http://ldapman.org/articles/intro_to_ldap.html, http://ldapman.org/servers.html, http://linuxplanet.com/linuxplanet/tutorials/5050/1/, and http://linuxplanet.com/linuxplanet/tutorials/5074/4/.

10.1.5 RADIUS AND TACACS USER CONFIGURATION

Users may be added to the local console server appliance. If they are not added and they log in via remote AAA, a user will be added for them. This user will not show up in the console server configurators unless they are specifically added, at which point they are transformed into a completely local user. The newly added user must authenticate off of the remote AAA server, and will have no access if it is down.

If a local user logs in, they may be authenticated or authorized from the remote AAA server, depending on the chosen priority of the remote AAA. A local user's authorization is the union of local and remote privileges.

EXAMPLE 1

User Tim is locally added, and has access to ports 1 and 2. He is also defined on a remote TACACS server, which says he has access to ports 3 and 4. Tim may log in with either his local or TACACS password, and will have access to ports 1 through 4. If TACACS is down, he will need to use his local password, and will only be able to access ports 1 and 2.

EXAMPLE 2

User Ben is only defined on the TACACS server, which says he has access to ports 5 and 6. When he attempts to log in a new user will be created for him, and he will be able to access ports 5 and 6. If the TACACS server is down he will have no access.

EXAMPLE 3

User Paul is defined on a RADIUS server only. He has access to all serial ports and network hosts.

EXAMPLE 4

User Don is locally defined on an appliance using RADIUS for AAA. Even if Don is also defined on the RADIUS server he will only have access to those serial ports and network hosts he has been authorized to use on the appliance.

If a no local AAA option is selected, then root will still be authenticated locally.

Remote users may be added to the admin group via either RADIUS or TACACS. Users may have a set of authorizations set on the remote TACACS server. Users automatically added by RADIUS will have authorization for all resources, whereas those added locally will still need their authorizations specified.

LDAP has not been modified, and will still need locally defined users.

NOTE: To interact with RADIUS, TACACS+ and LDAP with console server firmware v2.4.2 and earlier, user accounts on the local console server must also be set up. All resource authorizations must be added to the local appliance. With this release, if remote AAA is selected, it is used for password checking only. Root is always authenticated locally. Changes to PAM configurations will be destroyed next time the authentication configurator is run.

10.1.6 GROUP SUPPORT WITH REMOTE AUTHENTICATION

All console servers allow remote authentication via RADIUS, LDAP and TACACS+. With firmware v3.2 and later, RADIUS and LDAP can provide additional restrictions on user access based on group information or membership. For example, with remote group support, users can belong to a local group that has been set up to have restricted access to serial ports, network hosts and managed devices.

Remote authentication with group support works by matching a local group name with a remote group name provided by the authentication service. If the list of remote group names returned by the authentication service matches any local group names, the user is given permissions as configured in the local groups.

To enable group support to be used by remote authentication services:

- Navigate to Serial & Network > Authentication.

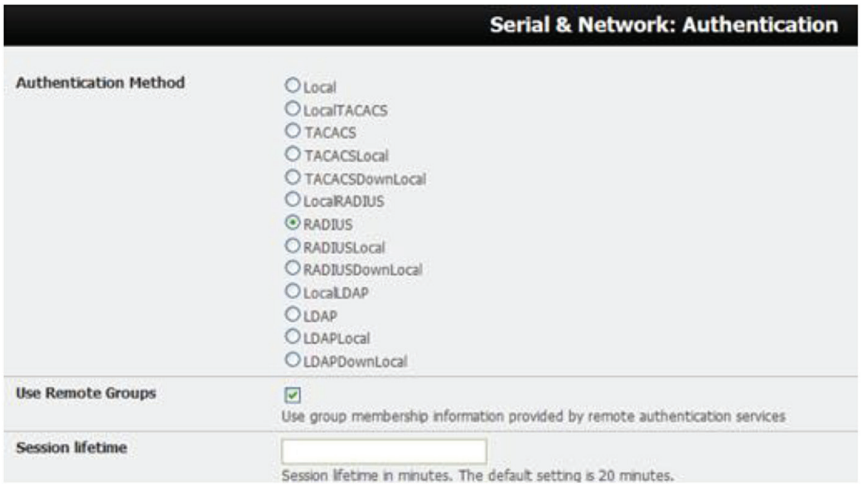


FIGURE 10-5.

- Select the relevant Authentication Method.
- Check the Use Remote Groups checkbox.

10.1.7 REMOTE GROUPS WITH RADIUS AUTHENTICATION

- Enter the RADIUS Authentication and Authorization Server Address and Server Password.
- Click Apply.
- Edit the Radius user's file to include group information and restart the Radius server.

When using RADIUS authentication, group names are provided to the console server using the Framed-Filter-Id attribute. This is a standard RADIUS attribute, and may be used by other devices that authenticate via RADIUS.

To interoperate with other devices using this field, the group names can be added to the end of any existing content in the attribute, in the following format:

:group_name=testgroup1,users:

This example sets the remote user as a member of testgroup1 and users, if these groups exist on the console server. Groups that do not exist on the console server are ignored.

FIGURE 10-6.

When setting the Framed-Filter-Id, the system may also remove the leading colon for an empty field. To work around this, add some dummy text to the start of the string. For example:

dummy:group_name=testgroup1,users:

If no group is specified for a user—for example AmandaJones—then the user will have limited console access, with no user interface or serial port access.

Default groups available on the console server include admin for administrator access and users for general user access.

TomFraser	Cleartext-Password := "FraTom70"
	Framed-Filter-Id=":group_name=admin:"
AmandaJones	Cleartext-Password := "JonAma83"
FredWhite	Cleartext-Password := "WhiFre62"
	Framed-Filter-Id=":group_name=testgroup1,users:"
JanetLong	Cleartext-Password := "LonJan57"
	Framed-Filter-Id=":group_name=admin:"

Additional local groups such as testgroup1 can be added via Users & Groups > Serial & Network.

10.1.8 REMOTE GROUPS WITH LDAP AUTHENTICATION

Unlike RADIUS, LDAP has built in support for group provisioning, which makes setting up remote groups easier. The console server will retrieve a list of all the remote groups that the user is a direct member of, and compare their names with local groups on the console server.

NOTE: Spaces in an LDAP group name will be converted to underscores.

For example, in an existing Active Directory setup, a group of users may be part of the UPS Admin and Router Admin groups.

On the console server, these users will be required to have access to a group Router_Admin, with access to port 1 (connected to the router), and another group, UPS_Admin, with access to port 2 (connected to the UPS).

Once LDAP is setup, users that are members of each group will have the appropriate permissions to access the router and UPS.

Currently, the only LDAP directory service that supports group provisioning is Microsoft Active Directory. Support is planned for OpenLDAP at a later time.

To enable group information to be used with an LDAP server:

- Complete the fields for standard LDAP authentication including LDAP Server Address, Server Password, LDAP Base DN, LDAP Bind DN and LDAP User Name Attribute.
- Enter memberOf for LDAP Group Membership Attribute as group membership is currently only supported on Active Directory servers.
- If required, enter the group information for LDAP Console Server Group DN, LDAP Administration Group DN, or both.

  A user must be a member of the LDAP Console Server Group DN group to gain access to the console and user interface. For example, the user must be a member of MyGroup on the Active Server to gain access to the console server.

  Additionally, a user must be a member of the LDAP Administration Group DN to gain administrator access to the console server. For example, the user must be a member of AdminGroup on the Active Server to receive administration privileges on the console server.

FIGURE 10-7.

- Click Apply.
- Ensure the LDAP service is operational and group names are correct within the Active Directory.

NOTE: When you are using remote groups with LDAP remote auth, you need to have corresponding local groups on the console server. Where the LDAP group names can contain upper case and space characters, the local group name on the console server must be all lower case and the spaces replaced with underscrores. For example, a remote group on the LDAP server may be My Ldap Access Group. The corresponding local group on the console server must be my_ldap_access_group. The local group on the console server must specify what the group member is granted access to for any group membership to be effective.

10.1.9 REMOTE GROUPS WITH TACACS+ AUTHENTICATION

When using TACACS+ authentication, there are two ways to grant a remotely authenticated user privileges. The first is to set the priv-lvl and port attributes of the raccess service to 12. See Section 10.2 for more information.

Additionally, or alternatively, group names can be provided to the console server using the groupname custom attribute of the raccess service.

An example Linux tac-plus config snippet might look like:

```
user = myuser {
  service = raccess {
    groupname="users"
    groupname1="routers"
    groupname2="dracs"
  }
}
```

You may also specify multiple groups in one comma-delimited. For example:

```
groupname="users,routers,dracs"
```

NOTE: The maximum length of the attribute value string is 255 characters.

To use an attribute name other than "groupname", set the Authentication > TACACS+ > TACACS Group Membership Attribute.

10.1.10 IDLE TIMEOUT

You can specify the time the console server waits before it terminates an idle ssh, pmshell or web connection.

- Navigate to Serial & Network > Authentication.

FIGURE 10-8.

- Set a Web Management Session Timeout in minutes. This specifies the browser console session idle timeout. The default setting is 20 minutes.

- Set a CLI Management Session Timeout in minutes. This specifies the ssh console session idle timeout. The default setting is to never expire.
- Set a Console Server Session Timeout in minutes.

  This specifies the pmshell serial console server session idle timeout. The default setting is to never expire.

10.1.11 KERBEROS AUTHENTICATION

Kerberos authentication can be used with UNIX and Windows (Active Directory) Kerberos servers. This form of authentication does not provide group information, so a local user with the same username must be created, and permissions set.

NOTE: Kerberos is sensitive to time differences between the Key Distribution Center (KDC) authentication server and the client device. Make sure that NTP is enabled, and the time zone is set correctly on the console server.

FIGURE 10-9.

When authenticating against Active Directory, the Kerberos Realm will be the domain name, and the Master KDC will be the address of the primary domain controller.

10.1.12 KERBEROS AUTHENTICATION

Console servers running firmware V3.5.2u3 or later include the Serial & Network > Authentication > Authentication Testing tab. This tab enables the connection to the remote authentication server to be tested.

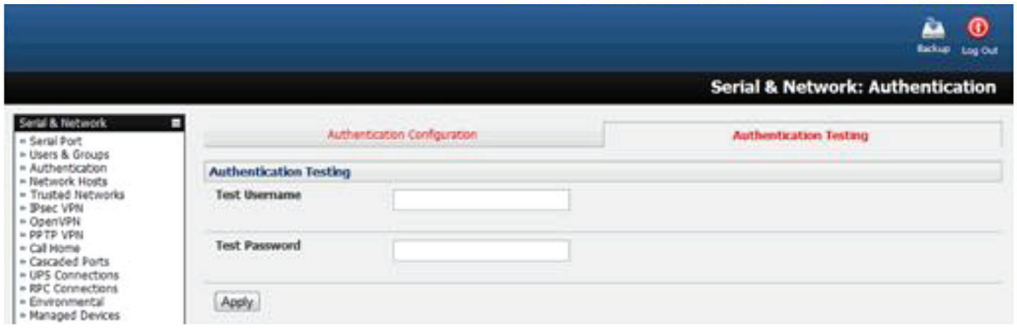


FIGURE 10-10.

10.2 PLUGGABLE AUTHENTICATION MODULES (PAM)

Console servers support RADIUS, TACACS+ and LDAP for two-factor authentication via PAM (Pluggable Authentication Modules). PAM is a flexible mechanism for authenticating users. A number of new ways of authenticating users have become popular. The challenge is that each time a new authentication scheme is developed; it requires all the necessary programs (login, ftpd etc.) to be rewritten to support it.

PAM provides a way to develop programs that are independent of authentication scheme. These programs need "authentication modules" attached to them at run-time to work. Which authentication module is attached depends on the local system setup and is at the discretion of the local Administrator.

The console server family supports PAM to which we have added the following modules for remote authentication:

TABLE 10-1. PAM MODULES

MODULE	BINARY	SOURCE
RADIUS	pam_radius_auth	https://freeradius.org/pam_radius_auth/
TACACS+	pam_tacplus	https://github.com/jeroennijhof/pam_tacplus
LDAP	pam_ldap	http://padl.com/OSS/pam_ldap.html

Further modules can be added as required.

Changes made to files in /etc/config/pam.d/ will persist, even if the authentication configurator is run.

- Users added on demand. When a user attempts to log in, but does not already have an account on the console server, a new user account will be created. This account will have no rights, and no password set. They will not appear in the Black Box configuration tools. Automatically added accounts will not be able to log in if the remote servers are unavailable.
- Admin rights granted over AAA. Users may be granted Administrator rights via networked AAA.

  For TACACS a priv-lvl of 12 or above indicates an administrator.

  For RADIUS, administrators are indicated via the Framed Filter ID. See the example configuration files below for example.
- Authorization via TACACS, LDAP or RADIUS for using remote groups. See Section 10.1.6.
- Authorization via TACACS for both serial ports and host access.

  Permission to access resources may be granted via TACACS by indicating a Black Box Appliance and a port or networked host the user may access. See the example configuration files below for example.

TACACS example

```
user = tim {
   service = raccess {
   priv-lvl = 11
   port1 = LES1604A/port02
 }
 global = cleartext mit
}
```

RADIUS example

```
paul Cleartext-Password: = "luap"
    Service-Type = Framed-User,
```

Fall-Through = No,

Framed-Filter-Id =":group_name=admin:"

The list of groups may include any number of entries separated by a comma. If the admin group is included, the user will be made an Administrator.

If there is already a Framed-Filter-Id, add the list of group_names after the existing entries, including the separating colon :.

10.3 SSL CERTIFICATE

The console server uses the Secure Socket Layer (SSL) protocol for encrypted network traffic between itself and a connected user.

During connection establishment, the console server has to expose its identity to the user's browser using a cryptographic certificate. The default certificate that comes with the console server device upon delivery is for testing purposes only and should not be relied on for secure global access.

NOTE: System administrators must not rely on the default certificate as the secured global access mechanism for use through Internet.

- Switch to your preferred browser.
- Enter https://ip-address-or-hostname-of-console-server-here/.

  Your browser may respond with a message that verifies the security certificate is valid but notes that it is not necessarily verified by a certifying authority.
- To proceed, you need to click yes if you are using Internet Explorer or select accept this certificate permanently (or temporarily) if you are using Mozilla Firefox.
- The Management Console login will present.
- Enter an Administrator's username and password as normal.

NOTE: We recommend that you generate and install a new base64 X.509 certificate that is unique for each particular console server.

To generate a new base64 X.509 certificate, the console server must be enabled to generate a new cryptographic key and the associated Certificate Signing Request (CSR) that needs to be certified by a Certification Authority (CA).

A certification authority verifies that you are the person who you claim you are, and signs and issues a SSL certificate to you. To create and install a SSL certificate for the console server:

- Navigate to System > SSL Certificate.
- Fill out the presented fields.

  Common name: the network name of the console server once it is installed in the network. Usually the fully qualified domain name. It is identical to the name used to access the console server with a web browser (without the "http://" prefix). If the name given here and the actual network name differ, the browser will pop up a security warning when the console server is accessed using https.

  Organizational Unit: this field is used for specifying to which department within an organization the console server belongs.

  Organization: the name of the organization to which the console server belongs.

  Locality/City: the city where the organization is located.

  State/Province: the state or province where the organization is located.

  Country: the two-letter ISO code designating the country where the organization is located.

  For example, DE for Germany and US for the the United States of America.

NOTE: The country code must be entered in ALL CAPS.

  Email: the email address of the person responsible for the console server and its security.

  Challenge Password: some certification authorities require a challenge password to authorize later changes on the certificate (for example, revocation of the certificate).

  Confirm Challenge Password: confirmation of the Challenge Password.

Key length: this is the length of the generated key in bits. 1024 Bits are supposed to be sufficient for most cases. Longer keys may result in slower response time of the console server during connection establishment.

- Click Generate CSR.

  The Certificate Signing Request (CSR) generation is initiated.

- Click Download to copy the CSR to your administration machine.
- Send the saved CSR string to a Certification Authority (CA) for certification.

  You will get the new certificate from the CA after a more or less complicated traditional authentication process (depending on the CA).

- Upload the certificate received from your CA to the console server using the Upload button.

After completing these steps the console server has its own certificate that is used for identifying the console server to its users.

NOTE: Information on issuing certificates and configuring HTTPS from the command line can be found in Chapter 16.

CHAPTER 11: NAGIOS INTEGRATION

Nagios is a powerful, highly extensible open source tool for monitoring network hosts and services. The core Nagios software package will typically be installed on a server or virtual server, the central Nagios server.

Console servers operate in conjunction with a central/upstream Nagios server to provide distributing monitoring of attached network hosts and serial devices. They embed the NSCA (Nagios Service Checks Acceptor) and NRPE (Nagios Remote Plug-in Executor) add-ons—this allows them to communicate with the central Nagios server, eliminating the need for a dedicated slave Nagios server at remote sites.

The console servers all support distributed monitoring. Even if distributed monitoring is not required, the Console servers can be deployed locally alongside the Nagios monitoring host server, to provide additional diagnostics and points of access to managed devices.

NOTE: If you have an existing Nagios deployment, you may wish to use the console server gateways in a distributed monitoring server capacity only. If this case and you are already familiar with Nagios, skip ahead to Section 11.3.

11.1 NAGIOS OVERVIEW

Nagios provides central monitoring of the hosts and services in your distributed network. Nagios is freely downloadable, open source software. This section offers a quick background of Nagios and its capabilities. A complete overview, FAQ and comprehensive documentation is available at https://nagios.org/.

Nagios forms the core of many leading commercial system management solutions such as GroundWork, https://gwos.com/.

Nagios does take some time to install and configure. Once it is up and running, it provides an outstanding network monitoring system. With Nagios you can:

- Display tables showing the status of each monitored server and network service in real time.
- Use a wide range of freely available plug-ins to make detailed checks of specific services. For example, don't just check a database is accepting network connections, check that it can actually validate requests and return real data.
- Display warnings and send warning e-mails, pager or SMS alerts when a service failure or degradation is detected.
- Assign contact groups who are responsible for specific services in specific time frames.

11.2 CONFIGURING NAGIOS DISTRIBUTED MONITORING

To activate the console server Nagios distributed monitoring:

Nagios integration must be enabled and a path established to the central/upstream Nagios server.

If the console server is to periodically report on Nagios monitored services, then the NSCA client embedded in the console server must be configured. The NSCA program enables scheduled check-ins with the remote Nagios server and is used to send passive check results across the network to the remote server.

If the Nagios server is to actively request status updates from the console server, then the NRPE server embedded in the console server must be configured. The NRPE server is the Nagios daemon for executing plug-ins on remote hosts.

Each of the Serial Ports and each of the Hosts connected to the console server that are to be monitored must have Nagios enabled and any specific Nagios checks configured.

The central/upstream Nagios monitoring host must be configured.

11.2.1 ENABLE NAGIOS ON THE CONSOLE SERVER

- Navigate to System > Nagios.

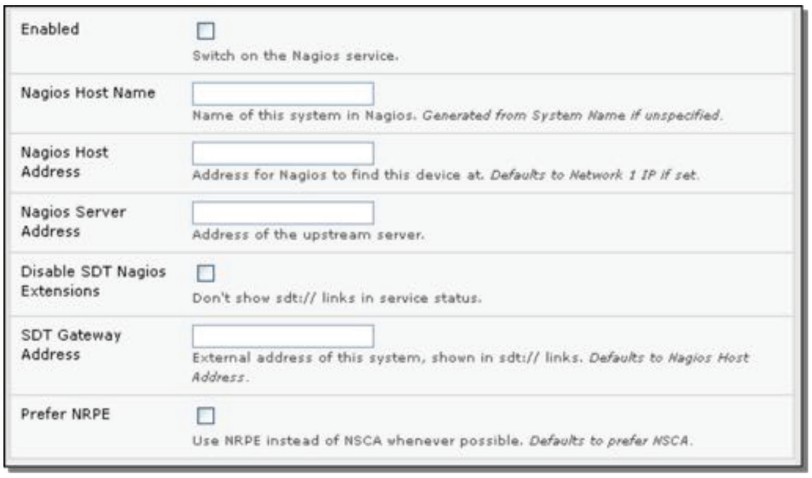


FIGURE 11-1.

- Check Enabled.
- Enter the Nagios Host Name the console server will be referred to in the Nagios server.

  This is generated from the local System Name (System > Administration) if unspecified.
- In Nagios Host Address, enter the address or hostname the upstream Nagios server uses to reach the console server. This defaults to the 1st network port: Network (1) (System > IP).
- In Nagios Server Address, enter the IP address or DNS name that the console server will use to reach the upstream Nagios monitoring server.
- Check the Disable SDT Nagios Extensions option to disable SDT Connector integration with your Nagios server at the head end. Only check to run vanilla Nagios monitoring.
- If not, enter the IP address or DNS name the SDT Nagios clients will use to reach the console server in SDT Gateway Address.
- Check Prefer NRPE to use NRPE when possible (for all communication except alerts).

  When NRPE and NSCA are both enabled, NSCA is the preferred method for communicating with the upstream Nagios server.

11.2.2 ENABLE NRPE MONITORING

Enabling NRPE allows you to execute plug-ins (such as check_tcp and check_ping) on the remote console server to monitor serial or network attached remote servers.

This will offload CPU load from the upstream Nagios monitoring machine that is especially valuable if you are monitoring hundreds or thousands of hosts.

To enable NRPE:

- Select System > Nagios.

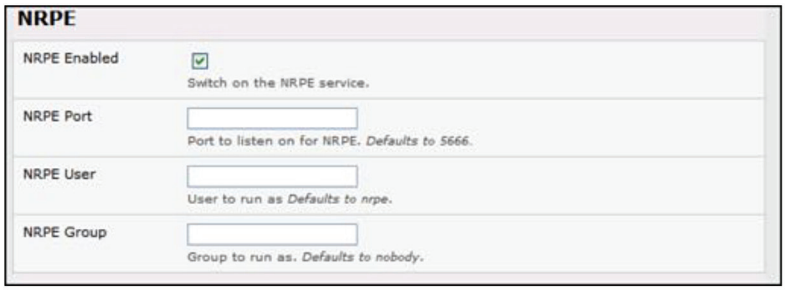


FIGURE 11-2.

- Check NRPE Enabled.
- Enter the details of the user connection to the upstream Nagios monitoring server.

  Refer to the sample Nagios configuration example for details of configuring specific NRPE checks.

By default, the console server will accept a connection between the upstream Nagios monitoring server and the NRPE server with SSL encryption, without SSL, or tunneled through SSH. The security for the connection is configured at the Nagios server.

11.2.3 ENABLE NSCA MONITORING

NSCA is the mechanism that allows you to send passive check results from the remote console server to the Nagios daemon running on the monitoring server.

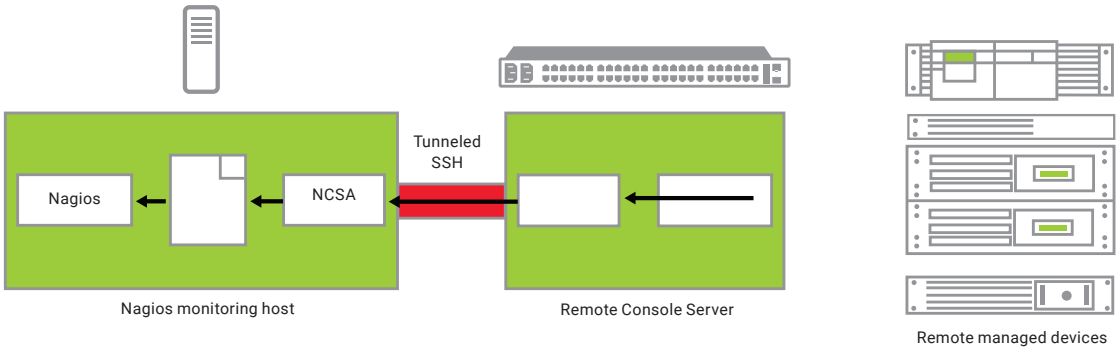


FIGURE 11-3.

To enable NSCA:

- Navigate to System > Nagios.
- Check the NSCA Enabled checkbox.
- Select the NSCA Encryption to be used from the drop-down menu.
- Enter an NSCA Secret password.
- Specify an NSCA Interval (in minutes).

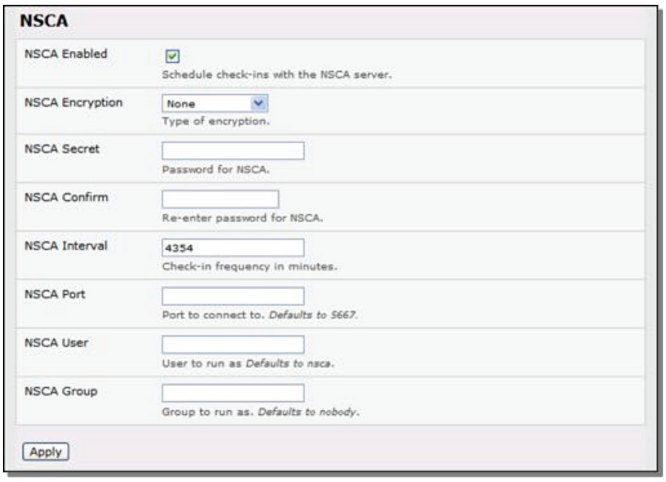


FIGURE 11-4.

For more on configuring specific NSCA checks, see the sample Nagios configuration described next.

11.2.4 CONFIGURE SELECTED SERIAL PORTS FOR NAGIOS MONITORING

The individual serial ports connected to the console server to be monitored must be configured for Nagios checks. See Section 5.4 for details on enabling Nagios monitoring for Hosts that are network connected to the console server.

To enable Nagios to monitor on a device connected to the console server serial port:

- Navigate to Serial & Network > Serial Port.

FIGURE 11-5.

- Click Edit on the serial Port # to be monitored.
- Check the Enable Nagios checkbox.
- Specify the Host Name of the device on the upstream server.
- Check the checkboxes of the Nagios checks to be run on this port.

  Serial Status monitors the handshaking lines on the serial port. Check Port monitors the data logged for the serial port.

11.2.5 CONFIGURE SELECTED NETWORK PORTS FOR NAGIOS MONITORING

The individual network hosts connected to the console server to be monitored must also be configured for Nagios checks.

- Navigate to Serial & Network > Network Port.

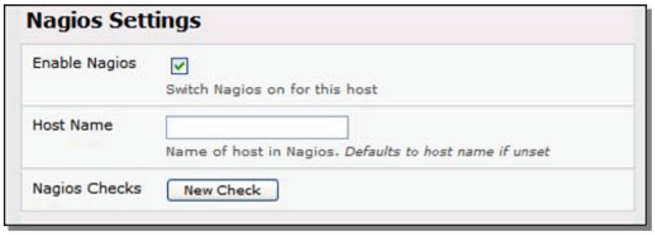


FIGURE 11-6.

- Click Edit on the Network Host to be monitored.
- Check the Enable Nagios checkbox.
- Specify the Host Name of the device as it will appear on the upstream Nagios server.
- Click New Check to add a specific check which will be run on this host.
- Select Check Permitted TCP or Check Permitted UDP to monitor a service that you have previously added as a permitted service.
- Alternatively, select Check TCP or Check UDP to specify a service port that you wish to monitor, but do not wish to allow external (SDT Connector) access to.
- The Nagios Check nominated as the check-host-alive check is the check used to determine whether the network host itself is up or down.

  Typically, this will be Check Ping, although in some cases the host will be configured not to respond to pings.

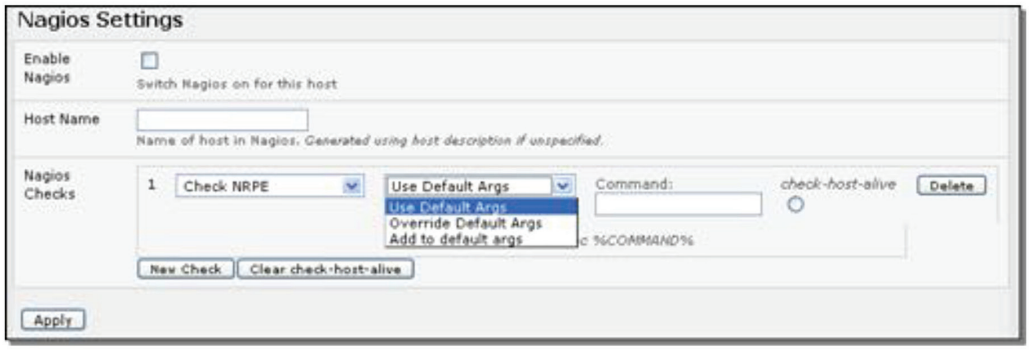


FIGURE 11-7.

- You can deselect check-host-alive.

  If check-host-alive check is de-selected, the host will always be assumed to be up.
- If required, customize the selected Nagios Checks to use custom arguments.
- Click Apply.

11.2.6 CONFIGURE THE UPSTREAM NAGIOS MONITORING HOST

For configuring the upstream server refer to the Nagios documentation, at https://nagios.org/documentation/.

The section entitled Distributed Monitoring steps through what you need to do to configure NSCA on the upstream server (under Central Server Configuration).

NRPE Documentation has recently been added which steps through configuring NRPE on the upstream server http://nagios.sourceforge.net/docs/nrpe/NRPE.pdf

At this stage, Nagios at the upstream monitoring server has been configured, and individual serial port and network host connections on the console server configured for Nagios monitoring. If NSCA is enabled, each selected check will be executed once over the period of the check interval. If NRPE is enabled, then the upstream server will be able to request status updates under its own scheduling.

11.3 ADVANCED DISTRIBUTED MONITORING CONFIGURATION

11.3.1 SAMPLE NAGIOS CONFIGURATION

An example configuration for Nagios is listed below. It shows how to set up a remote console server to monitor a single host, with both network and serial connections.

For each check it has two configurations: one for NRPE and one for NSCA.

In practice, these would be combined into a single check which used NSCA as a primary method, falling back to NRPE if a check was late. For details, see the Nagios documentation —at https://nagios.org/documentation/—on Service and Host Freshness Checks.

```
; Host definitions
; Black Box console server
define host {
  use                 generic-host
  host_name               Black Box
  alias               Console server
  address               192.168.254.147
}
; Managed Host
define host {
  use                 generic-host
  host_name               server
  alias               server
  address               192.168.254.227
}
; NRPE daemon on gateway
define command {
  command_name            check_nrpe_daemon
```

```
  command_line                $USER1$/check_nrpe -H \
                        192.168.254.147 -p 5666
}
define service {
  service_description          NRPE Daemon
  host_name                    Black Box
  use                          generic-service
  check_command                check_nrpe_daemon
}
; Serial Status
define command {
  command_name                 check_serial_status
  command_line                 $USER1$/check_nrpe -H \
                        192.168.254.147 -p 5666 -c \
                        check_serial_$HOSTNAME$
}
define service {
  service_description          Serial Status
  host_name                    server
  use                          generic-service
  check_command                check_serial_status
}
define service {
  service_description          serial-signals-server
  host_name                    server
  use                          generic-service
  check_command                check_serial_status
  active_checks_enabled        0
  passive_checks_enabled       1
}
define servicedependency {
  name                         Black Box_nrpe_daemon_dep
  host_name                    Black Box
  dependent_host_name          server
  dependent_service_description Serial Status
  service_description          NRPE Daemon
  execution_failure_criteria   w,u,c
}
```

```
; Port Log
define command {
 command_name              check_port_log
 command_line              $USER1$/check_nrpe -H \
                    192.168.254.147 -p 5666 -c \
                    port_log_$HOSTNAME$
}
define service {
 service_description       Port Log
 host_name                 server
 use                    generic-service
 check_command             check_port_log
}
define service {
 service_description       port-log-server
 host_name                 server
 use                    generic-service
 check_command             check_port_log
 active_checks_enabled     0
 passive_checks_enabled    1
}
define servicedependency {
 name                   Black Box_nrpe_daemon_dep
 host_name              Black Box
 dependent_host_name       server
 dependent_service_description Port Log
 service_description       NRPE Daemon
 execution_failure_criteria  w,u,c
}
; Ping
define command {
 command_name              check_ping_via_Black Box
 command_line              $USER1$/check_nrpe -H \
                    192.168.254.147 -p 5666 -c \
                    host_ping_$HOSTNAME$
}
define service {
 service_description       Host Ping
 host_name                 server
```

```
  use                     generic-service
  check_command               check_ping_via_Black Box
}
define service {
  service_description         host-ping-server
  host_name                 server
  use                     generic-service
  check_command               check_ping_via_Black Box
  active_checks_enabled       0
  passive_checks_enabled      1
}
define servicedependency {
  name                    Black Box_nrpe_daemon_dep
  host_name                 Black Box
  dependent_host_name         server
  dependent_service_description Host Ping
  service_description         NRPE Daemon
  execution_failure_criteria    w,u,c
}
; SSH Port
define command {
  command_name              check_conn_via_Black Box
  command_line              $USER1$/check_nrpe -H \
                        192.168.254.147 -p 5666 -c \
                        host_$HOSTNAME$_$ARG1$_$ARG2$
}
define service {
  service_description         SSH Port
  host_name                 server
  use                     generic-service
  check_command               check_conn_via_Black Box!tcp!22
}
define service {
  service_description         host-port-tcp-22-server
                      ; host-port-<protocol>-<port>-<host>
  host_name                 server
  use                     generic-service
  check_command               check_conn_via_Black Box!tcp!22
  active_checks_enabled       0
```

```
  passive_checks_enabled       1
}
define servicedependency {
  name                         Black Box_nrpe_daemon_dep
  host_name                    Black Box
  dependent_host_name          server
  dependent_service_description SSH Port
  service_description          NRPE Daemon
  execution_failure_criteria   w,u,c
}
```

11.3.2 BASIC NAGIOS PLUG-INS

Plug-ins are compiled executables or scripts that can be scheduled to be run on the console server to status check a connected host or service. This status is communicated to the Nagios server, which uses the results to monitor the status of the network. Console servers are preconfigured with a selection of checks that are part of the Nagios plug-ins package.

TABLE 11-1. NAGIOS PLUG-INS

PLUG-IN	DESCRIPTION
check_tcp	Used to check open ports on network hosts.
check_udp	Used to check open ports on network hosts.
check_ping	Used to check network host availability.
check_nrpe	Used to execute arbitrary plug-ins in other devices.
check_serial_signals	Used to monitor handshaking lines on serial ports. Black Box-specific.
check_port_log	Used to monitor the data logged for a serial port. Black Box-specific.

11.3.3 ADDITIONAL PLUG-INS

Additional Nagios plug-ins (listed below) are available for all LES1516A, LES1532A, LES1548A-, LES1700-R2- and LES1400-series devices.

TABLE 11-2. ADDITIONAL PLUG-INS

PLUG-IN	PLUG-IN	PLUG-IN	PLUG-IN	PLUG-IN	PLUG-IN
check_apt	check_by_ssh	check_clamd	check_apt	check_by_ssh	check_clamd
check_dig	check_dns	check_dummy	check_fping	check_ftp	check_game
check_hpjd	check_http	check_imap	check_jabber	check_ldap	check_load
check_mrtg	check_mrtgtraf	check_nagios	check_nntp	check_nntps	check_nt
check_ntp	check_nwstat	check_overcr	check_ping	check_pop	check_procs
check_real	check_simap	check_smtp	check_snmp	check_spop	check_ssh
check_ssmtp	check_swap	check_tcp	check_time	check_udp	check_ups
check_users	—	—	—	—	—

To get these plug-ins from the Nagios plug-ins package, contact Black Box Technical Support at 877-877-2269 or info@blackbox.com

To configure additional checks, the downloaded plug-in program must be saved in the tftp addins directory on the USB flash drive and the downloaded text plug-in file saved in /etc/config/.

To enable these additional plug-ins:

- Navigate to Serial & Network > Network Port.
- Click Edit for the Network Host to be monitored.
- Select New Checks.

  The additional check options will have been included in the updated Nagios Checks list, and you can again customize the arguments.

If you need other plug-ins to be loaded into the LES1516A, LES1532A, LES1548A or LES1700-R2 firmware:

- If the plug-in is a Perl script, it must be rewritten. The console server does not support Perl at this point.
- Individual compiled programs may be generated using gcc for ARM.

11.3.4 NUMBER OF SUPPORTED DEVICES

Ultimately the number of devices that can be supported by any particular console server is a function of the number of checks being made, and how often they are performed. Access method will also play a part. The table below shows the performance of three of the console server models (1/2-port, 8-port and 16/48 port):

TABLE 11-3. NCSA TESTS

NCSA TESTS	NO ENCRYPTION	3DES	SSH
1 check	~ 0.5 sec	~ 0.5 sec	~ 0.5 sec
100 sequential checks	100.0 sec	100.0 sec	100.0 sec
10 sequential checks, batched upload	1.5 sec	2.0 sec	1.0 sec
100 sequential checks, batched upload	7.0 sec	11.0 sec	6.0 sec
NRPE TESTS	**NO ENCRYPTION**	**3DES**	**TUNNELED OVER SSH**
1 check	0.1 sec	0.3 sec	0.1 sec
10 simultaneous checks	1.0 sec	3.0 sec	1.3 sec
MAX. SIMULTANEOUS CHECKS BEFORE TIMEOUTS	**NO ENCRYPTION**	**3DES**	**SSH TUNNEL**
1-port and 2-port	30	20	25
8-port	30	20	25
16-port and 48-port	30	25	35

The results were from running tests 5 times in succession with no timeouts on any runs. However there are a number of ways to increase the number of checks you can do:

Usually when using NRPE checks, an individual request will need to set up and tear down an SSL connection. This overhead can be avoided by setting up an SSH session to the console server and tunneling the NRPE port. This allows the NRPE daemon to be run securely without SSL encryption, as SSH will take care of the security.

When the console server submits NSCA results it staggers them over a certain time period (e.g. 20 checks over 10 minutes will result in two check results every minute). Staggering the results like this means that in the event of a power failure or other incident that causes multiple problems, the individual freshness checks will be staggered too.

NSCA checks are also batched. So in the previous example the two checks per minute will be sent through in a single transaction.

11.3.5 DISTRIBUTED MONITORING USAGE SCENARIOS

Below are a number of distributed Nagios monitoring scenarios.

LOCAL OFFICE

In this scenario, the console server is set up to monitor the console of each managed device. It can be configured to make a number of checks, either actively at the Nagios server's request, or passively at preset intervals, and submit the results to the Nagios server in a batch.

The console server may be augmented at the local office site by one or more Intelligent Power Distribution Units (IPDUs) to remotely control the power supply to the managed devices.

REMOTE SITE

In this scenario, the console server NRPE server or NSCA client can be configured to make active checks of configured services and upload to the Nagios server waiting passively. It can also be configured to service NRPE commands to perform checks on demand.

In this situation, the console server will perform checks based on both serial and network access.

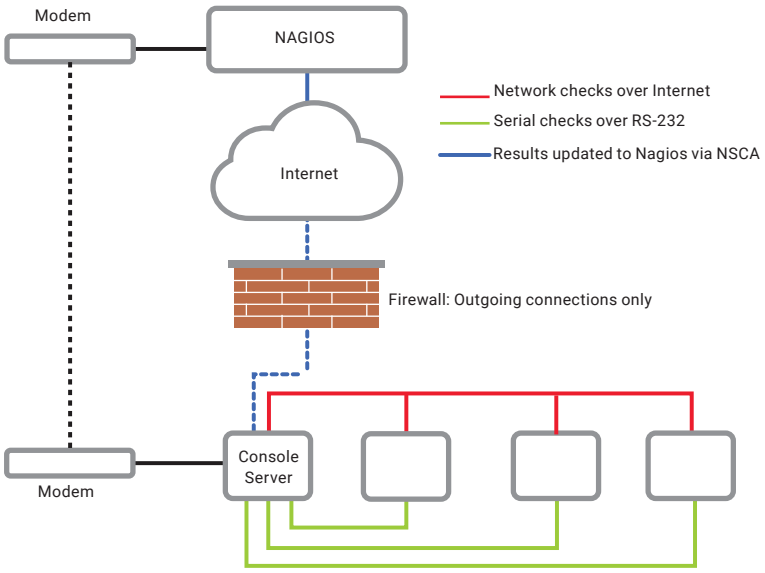


FIGURE 11-8.

REMOTE SITE WITH RESTRICTIVE FIREWALL

In this scenario, the role of the console server will vary. One aspect may be to upload check results through NSCA.

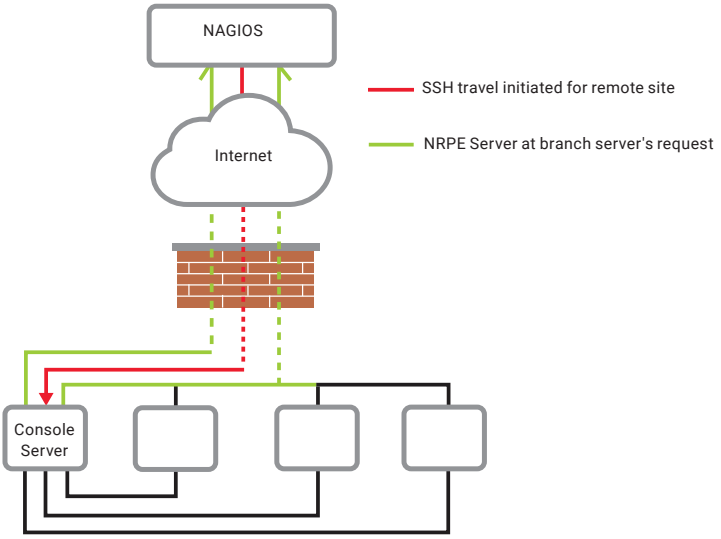


FIGURE 11-9.

Another may be to provide an SSH tunnel to allow the Nagios server to run NRPE commands.

REMOTE SITE WITH NO NETWORK ACCESS

In this scenario, the console server allows dial-in access for the Nagios server.

Periodically, the Nagios server will establish a connection to the console server and execute any NRPE commands, before dropping the connection.

CHAPTER 12: SYSTEM MANAGEMENT

This chapter documents how the Administrator can perform a range of general console server system administration and configuration tasks such as:

- Applying Soft and Hard Resets to the console server.
- Re-flashing the Firmware.
- Configuring the Date, Time and NTP.
- Setting up Backup of the configuration files.
- Delayed configuration commits.
- Configuring the console server in FIPS mode.

System administration and configuration tasks that are covered elsewhere include

- Section 4.2: Resetting the system.
- Section 4.3: Setting the console server's System IP Address.
- Section 4.4: Setting the Services permitted to access the console server.
- Chapter 6: Setting up OOB Dial-in.
- Chapter 13: Configuring the Dashboard.

12.1 SYSTEM ADMINISTRATION AND RESET

The Administrator can reboot or reset the gateway to default settings.

To effect a soft reset:

- Navigate to System > Administration
- Select Reboot.
- Click Apply.

The console server reboots with all settings (for example, the assigned network IP address) preserved. This soft reset disconnects all users and ends any SSH sessions that had been established.

A soft reset will also be effected when you switch OFF power from the console server, and then switch the power back ON.

NOTE: If you cycle the power and the unit is writing to flash you could corrupt or lose data. The software reboot is the safer option.

To effect a hard reset or hard erase:

- Push the Erase button on the rear panel gently twice within a few seconds period while the unit is powered on..

  A ball point pen or bent paper clip is a suitable tool for performing this procedure. Do not use a graphite pencil.

This will reset the console server back to its factory default settings and clear the console server's stored configuration information (for example, the unit's IP address will be reset to 192.168.0.1).

You will be prompted to log in and must enter the default administration username and password:

Username: root

Password: default

12.2 FIRMWARE UPGRADES

Before upgrading, you should ascertain if you are already running the most current firmware in your console server. Your console server will not allow you to upgrade to the same or an earlier version.

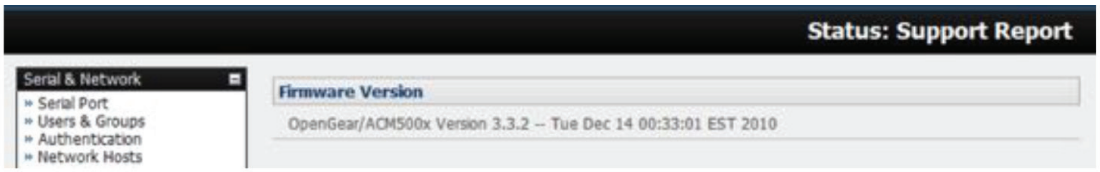


FIGURE 12-1.

- The Firmware version is displayed in the header of each page.
- Alternately selecting Status > Support Report reports the Firmware Version.
- To upgrade, contact Black Box Technical Support at 877-877-2269 or info@blackbox.com to get the latest firmware.
  - For LES1600 family, you will need LES1600.flash.
  - For LES1500 family, you will need LES1516A, LES1532A, LES1548A.flash.
  - For LES1708A-R2/16/32/48, you will need LES1700-R2.flash.
- Save the firmware image file onto a system on the same subnet as the Black Box device.

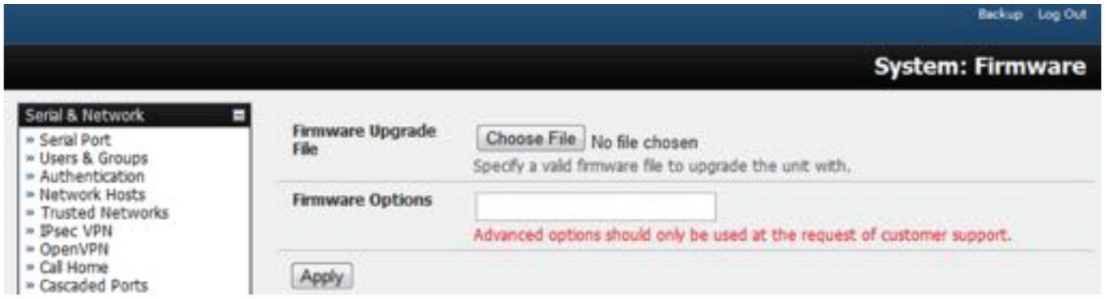


FIGURE 12-2.

- Select System > Firmware on the system to be upgraded.
- Specify the address and name of the downloaded firmware upgrade file, or browse the local subnet and locate the downloaded file.
- Click Apply.

  The Black Box device will undertake a soft reboot and commence upgrading the firmware. This process will take several minutes.
- After the firmware upgrade has completed, click here to return to the Management Console. Your Black Box device will have retained all its pre-upgrade configuration information.

12.3 DATE AND TIME CONFIGURATION

Set the local Date and Time in your Black Box appliance as soon as it is configured. Features such as Syslog and NFS logging use the system time for time-stamping log entries, while certificate generation depends on a correct Timestamp to check the validity period of the certificate.

Your Black Box appliance can synchronize its system time with a remote Network Time Protocol (NTP) server. NTP uses Coordinated Universal Time (UCT) for all time synchronizations so it is not affected by different time zones. You do need to specify your local time zone so the system clock shows correct local time.

- Select System > Date & Time.
- Set your appropriate region in the Time Zone selection box and click Set Time.

NOTE: With firmware v3.2.0 and later the Time Zone can also be set to UTC, which replaced Greenwich Mean Time as the World standard for time in 1986.

Configuring NTP ensures the Black Box appliance clock is kept extremely accurate once an Internet connection has been established.

- Select the Enable NTP checkbox in the Network Time Protocol section of the System > Date & Time page.
- Enter the IP address of the remote NTP Server.
- If your external NTP server requires authentication, specify the NTP Authentication Key and the Key Index to use when authenticating with the NTP server.

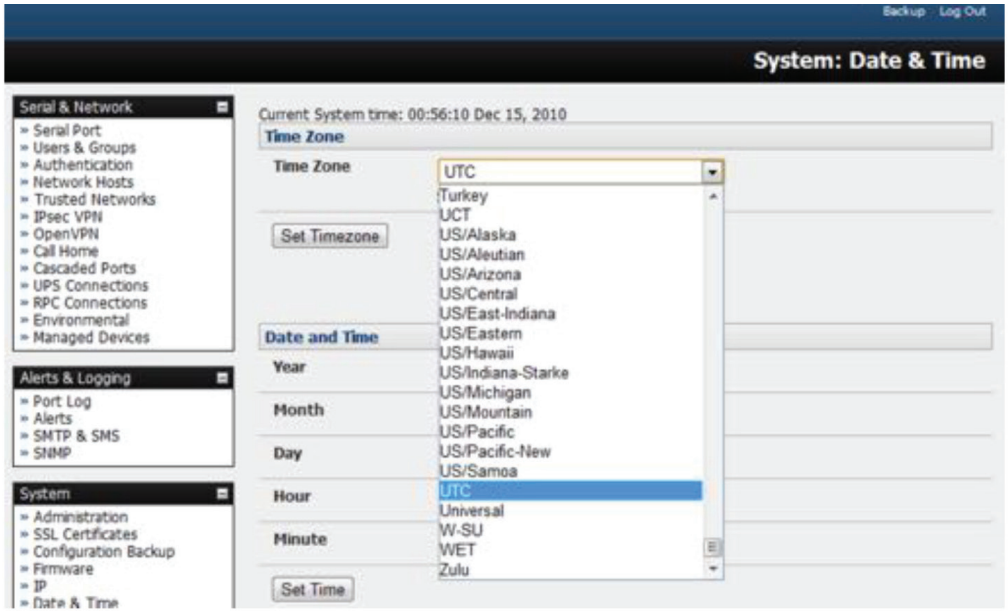


FIGURE 12-3.

- Click Apply NTP Settings.

If remote NTP is not used, the time can be set manually:

- Enter the Year, Month, Day, Hour and Minute using the Date and Time selection boxes.
- Check Set Time.

Black Box appliances have an internal, battery-backed hardware clock. Whether set manually, or set by an NTP server, the hardware clock is automatically updated. The clock's battery maintains the time and date across reboots and when the appliance is powered down.

With the NTP peering model, console servers can share time information with other connected devices, so all devices can be time synchronized. To do this, tick Enable NTP on the Time and Date page, and ensure the appropriate networks are selected on the Service Access page.

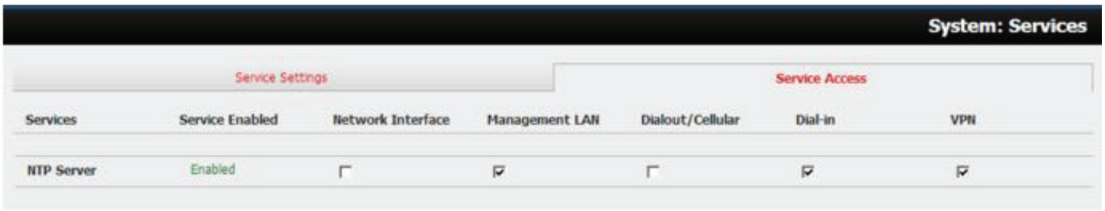


FIGURE 12-4.

12.4 BACKUP CONFIGURATION

We recommend that you back up the console server configuration whenever you make significant changes (such as adding new Users or Managed Devices) or before performing a firmware upgrade.

- Select System > Configuration Backup or click the Backup icon.

FIGURE 12-5. BACKUP ICON

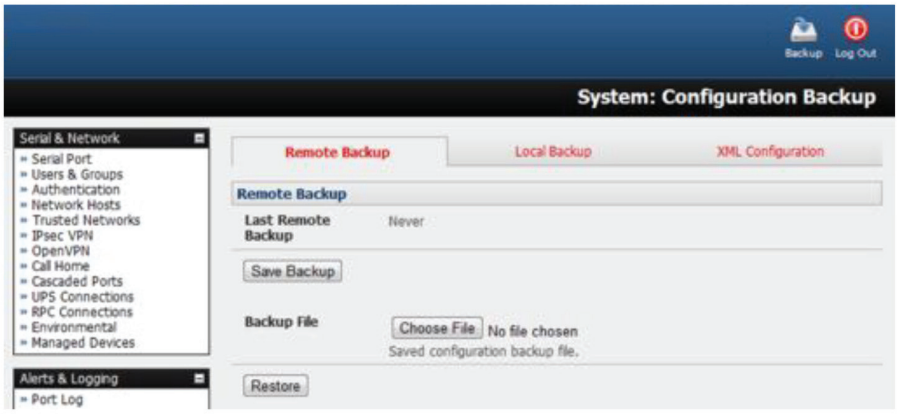


FIGURE 12-6. SYSTEM: CONFIGURATION BACKUP SCREEN

NOTE: Configuration files can also be backed up from the command line (see Chapter 15).

With all console servers, you can save the backup file remotely on your PC and you can restore configurations from remote locations:

- Navigate to System > Configuration Backup.
- Click the Remote Backup tab.

- Click Save Backup in the Remote Backup section.

  The config backup file—system-name_date_config.opg—will be downloaded to your PC and saved in the location you select.

To restore a remote backup:

- Navigate to System > Configuration Backup.
- Click the Remote Backup tab.
- Click Browse in the Remote Backup section.
- Select the backup file you wish to restore.
- Click Restore.
- Click OK.

NOTE: This will overwrite all the current configuration settings in your console server.

With some console servers, you can save the backup file locally onto the USB storage. To do this, your console server must support USB and you must have an internal or external USB flash drive installed.

To backup and restore using USB:

- Ensure the USB flash drive is the only USB device attached to the console server.
- Navigate to System > Configuration Backup.
- Select the Local Backup tab.
- Click click here to proceed.

  This will set a Volume Label on the USB storage device.

  This preparation step is only necessary the first time, and will not affect any other information you have saved onto the USB storage device. We recommend that you back up any critical data from the USB storage device before using it with your console server. If there are multiple USB devices installed, you will be warned to remove them.

FIGURE 12-7.

- To back up to the USB, enter a brief Description of the backup in the Local Backup menu and select Save Backup.
- The Local Backup menu will display all the configuration backup files you have stored onto the USB flash drive.
- To restore a backup from the drive, select Restore on the particular backup you wish to restore and click Apply.

After saving a local configuration backup, you may choose to use it as the alternate default configuration. When the console server is reset to factory defaults, it will then load your alternate default configuration instead of its factory settings.

To set an alternate default configuration:

- Check Load On Erase
- Click Apply.

NOTE: Before selecting Load On Erase, ensure you have tested your alternate default configuration by clicking Restore.

If your alternate default configuration causes the console server to become unbootable, recover your unit to factory settings.

If the configuration is stored on an external USB storage device, unplug the storage device and reset to factory defaults as per Section 12.1.

If the configuration is stored on an internal USB storage device, reset to factory defaults using a specially-prepared USB storage device.

This specially-prepared USB storage device:

- Must be formatted with a Windows FAT32/VFAT file system on the first partition or the entire disk.

  Most USB thumb drives are sold already formatted this way.
- The file system must have the volume label OPG_DEFAULT.
- Insert this USB storage device into an external USB port on the console server and reset to factory defaults as per Section 12.1.
- After recovering your console server, ensure the problematic configuration is no longer selected for Load On Erase.

12.5 DELAYED CONFIGURATION COMMIT

This mode allows the grouping or queuing of configuration changes and the simultaneous application of these changes to a specific device. For example, changes to authentication methods or user accounts may be grouped and run once to minimize system downtime. To enable:

- Navigate to System > Administration.
- Check the Delayed Config Commits checkbox.

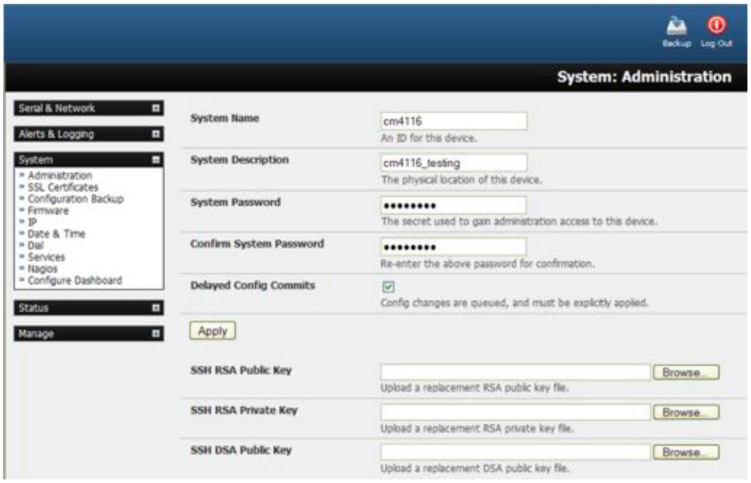


FIGURE 12-8.

- Click Apply.

The Commit Config icon will present in top right-hand corner of the screen between the Backup and Log Out icons.

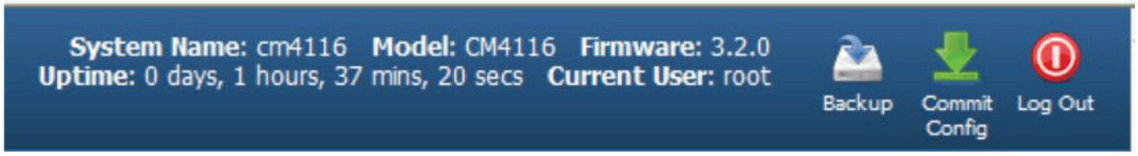


FIGURE 12-9.

To queue, then run, configuration changes:

- Apply all the required changes to the configuration.

  For example, modify user accounts, amend authentication method, enable OpenVPN tunnel or modify system time.
- Click the Commit Config button.

  This will generate the System > Commit Configuration screen displaying all the configurators to be run.

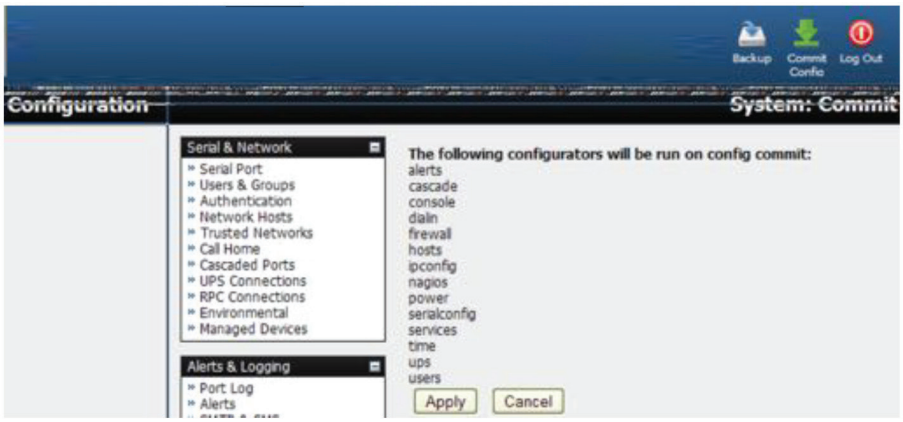


FIGURE 12-10.

- Click Apply.

  All the configurators in the queue will run.
- Alternatively, click Cancel.

  All the queued configuration changes will be lost.

To disable the Delayed Configuration Commits mode:

- Uncheck the Delayed Config Commits checkbox under System > Administration.
- Click Apply.
- Click the Commit Config button in top right-hand corner of the screen.

  The System > Commit Configuration screen displays.
- Click Apply.

  The systemsettings configurator runs

FIGURE 12-11.

The Commit Config button will no longer be displayed in the top right-hand corner of the screen and configurations will no longer be queued.

12.6 FIPS MODE

The LES1600, LES1508A, LES1200, LES1516A, LES1532A, LES1548A, LES1700-R2 and LES1400 family of advanced console servers
all use a FIPS 140-2 validated embedded cryptographic module.

NOTE: The US National Institute of Standards and Technology (NIST) publishes the FIPS (Federal Information Processing Standard) standards. FIPS 140-1 and FIPS 140-2 are both technical standards and worldwide de-facto standards for cryptographic module implementation. They are issued by NIST for use government-wide. NIST develops FIPS when there are government requirements, such as for security and interoperability, and no acceptable industry options. Black Box advanced console servers use an embedded OpenSSL crypto-graphic module validated to FIPS 140-2 standards and in receipt of Certificate #1051.

When configured in FIPs mode, all SSH, HTTPS and SDT Connector access to all services on the console server use the embedded FIPS-compliant module. To connect, your browser or client must also be using FIPs-approved cryptographic algorithms or the connection will fail.

- Select System > Administration.
- Check FIPS Mode

  This will enable FIPS mode after a safe reboot.

FIGURE 12-12.

- Check Reboot to safely reboot the console server.
- Click Apply.

  The console server reboots. It will take several minutes to reconnect as secure browser communications are validated.

  When reconnected it will display FIPs mode: Enabled in the Management Console banner.

To enable FIPS mode from the command line, login and run these commands:

config -s config.system.fips=on

touch /etc/config/FIPS

chmod 444 /etc/config/FIPS

flatfsd -b

The final command saves to flash and reboots the unit. The unit will take a few minutes to boot into FIPS mode.

To disable FIPS mode from the shell, run these commands:

config -d config.system.fips

rm /etc/config/FIPS

flatfsd –b

CHAPTER 13: STATUS REPORTS

This chapter documents the Dashboard feature and the status reports that are available:

- Port Access and Active Users
- Statistics.
- Support Reports.
- Syslog.
- Dashboard.

Other status reports that are covered elsewhere include:

- Section 9.1: RPC Status.
- Section 9.2: UPS Status.
- Section 9.3: Environmental Status.

13.1 PORT ACCESS AND ACTIVE USERS

The Administrator can see which Users have access privileges with which serial ports:

- Select Status > Port Access.

The Administrator can also see the current status of Users who have active sessions on those ports:

- Select Status > Active Users.

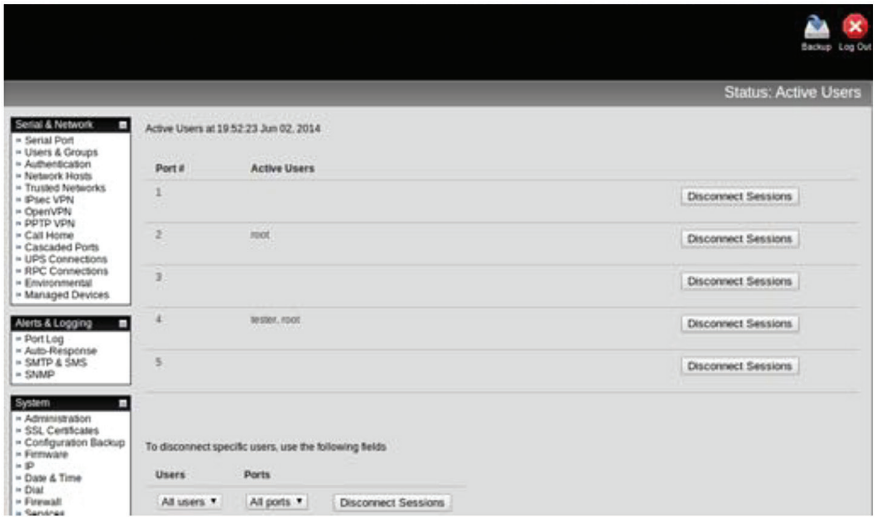


FIGURE 13-1.

With firmware v3.11 and later the Status > Active Users menu has been extended to enable Administrators to selectively terminate serial sessions. Connection types telnet, SSH, raw TCP and unauthenticated telnet can be disconnected. However you cannot disconnect an RFC2217 session.

The root user, or any user in the admin group, can access the Active Users page, which shows a snapshot of the connected sessions, at the time indicated by the timestamp displayed at the top of the page. Note that this page only shows the local console ports and does not include any cascaded ports.

There are Disconnect Sessions buttons along the right hand side of the table listing active users. These buttons disconnect all sessions from the Port they correspond to. If the port is not set up in Console Server mode, the user will see a pop up error informing them that they need to configure the port as Console Server mode before they can connect and disconnect.

After the buttons have been pressed, the selected sessions will be disconnected, and the number of disconnect sessions will be displayed to the user.

To allow more detailed control of who to disconnect, there is a table at the bottom of the page with drop-down lists for all connected users and all connected ports that allow the user to choose who do disconnect. If you wish to disconnect the user tester from all ports, choose tester in the Users box, and All ports in the Ports box then click Disconnect Sessions.

NOTE: You can also disconnect serial sessions from the command line using the --disconnect option with the pmusers command.

13.2 STATISTICS

The Statistics report provides a snapshot of the status, current traffic and other activities and operations of your console server.

- Select Status > Statistics.

Detailed statistics reports can be found by selecting the various tabs.

For example, if you have a console server configured with a wireless LAN connection the Wireless screen will display all the locally accessible wireless LANs.

FIGURE 13-2.

You can see the SSID and Encryption/Authentication settings for the desired access point.

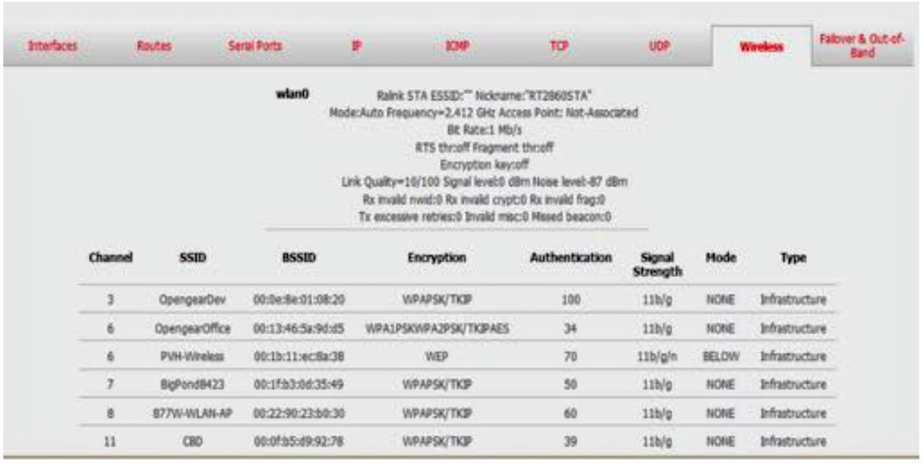


FIGURE 13-3.

Also when you have successfully connected, the SSID of this access point will then be shown in the Wireless ESSID filed of ra0 (shows above as "" which is not connected).

13.3 SUPPORT REPORTS

The Support Report provides status information that assists the Black Box technical support team to solve any problems you may experience with your console server. With email support requests, generate a Support Report when the issue is occurring, and attach it as text.

- Select Status > Support Report.

  A status snapshot presents.

- Save the file as a text file and attach it to your support email.

NOTE: For console servers running firmware v3.11 and above, for devices where the serial number can be retrieved, there is now a Feature Set section displaying the serial number. LES1200, LES1508A, LES1600, LES1516A, LES1532A, LES1548A and LES1700-R2 can display their serial number. For devices not supporting this feature, there is no change to the support report.

NOTE: There is also a new cli command on all devices called showserial that does nothing more than return the serial number if it is available or "No serial number information available." The command exists on all devices so VCMS bulk commands can be run on many console servers and obtain as many serial numbers as possible in one operation.

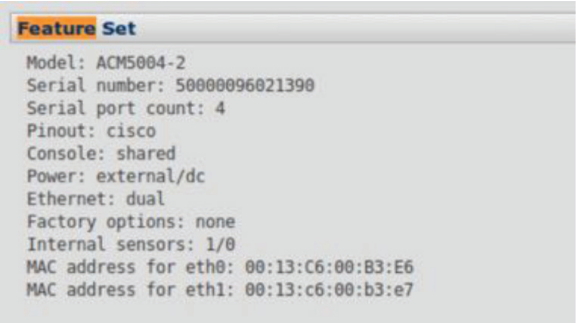


FIGURE 13-4.

13.4 SYSLOG

The console server's Linux system logger maintains a record of all system messages and errors:

- Select Status > Syslog.

The syslog record can be redirected to a remote Syslog Server:

- Enter the remote Syslog Server Address and Syslog Server Port details.
- Click Apply.

The console also maintains a local Syslog. To view this local Syslog file:

- Select Status > Syslog.

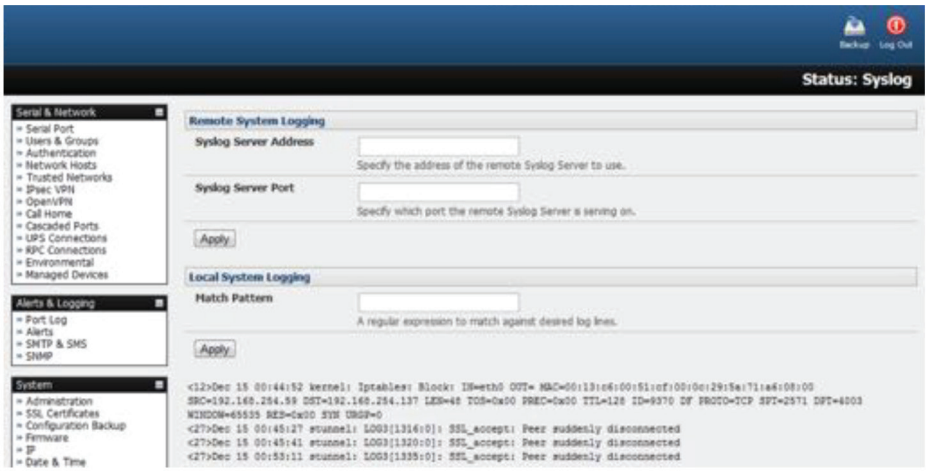


FIGURE 13-5.

To find specific information in the local Syslog file, a pattern matching filter tool is provided.

- Specify the Match Pattern that is to be searched for
- Click Apply.

  The Syslog will present with only those entries that include the specified pattern.

13.5 DASHBOARD

The Dashboard provides the administrator with a summary of the status of the console server and its Managed Devices.

Custom dashboards can be configured for each user group.

13.5.1 CONFIGURING THE DASHBOARD

Only the root user and users who are members of the admin group can configure and access the dashboard. To configure a custom dashboard:

- Select System > Configure Dashboard.
- Select the user (or group) you are configuring this custom dashboard layout for.

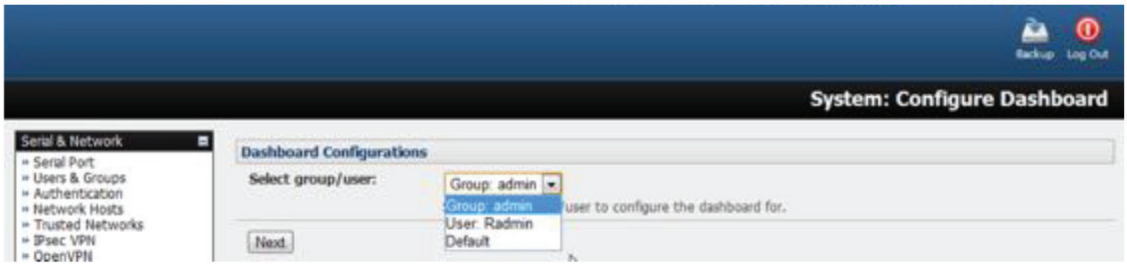


FIGURE 13-6

You can configure a custom dashboard for any admin user or for the admin group or you can reconfigure the default dashboard.

The Status > Dashboard screen is the first screen displayed when admin users (other than root) log into the console manager.

If you log in as john, and john is member of the admin group and there is a dashboard layout configured for john, then you will see the dashboard for john on log-in and each time you click on the Status > Dashboard menu item.

If there is no dashboard layout configured for john but there is an admin group dashboard configured, then you will see the admin group dashboard instead. If there is no user-specific dashboard or admin group dashboard configured, the default dashboard is displayed.

NOTE: The root user does not have its own dashboard.

The Dashboard displays a configurable number of widgets. These widgets include status for major subsystems such as conma, Auto-Response, Managed Devices and cellular.

The admin user can configure which of these widgets is to be displayed where:

- Go to the Dashboard layout panel and select the widgets to display in each Widget Slot.
- Click Apply.

NOTE: Dashboard configuration is stored in /etc/config/config.xml. Each configured dashboard will increase the size of this file. If this file gets too big, you can run out of memory space on the console server.

13.5.2 CREATING CUSTOM WIDGETS FOR THE DASHBOARD

To run a custom script in a dashboard widget, create a file called widget-<name>.sh in the folder /etc/config/scripts/. You can have as many custom dashboard files as you want.

Put any code inside this file. When configuring the dashboard, choose widget-<name>.sh from the dropdown list. The dashboard runs and displays the script's output inside the widget.

The best way to format the output is to send HTML back to the browser using echo:

```
echo '<table>'
```

You can run any command and its output will be displayed in the widget window directly.

Below is an example script. It writes the current date to a file, and then echo's HTML code back to the browser. The HTML code gets an image from a URL and displays it in the widget.

```
#!/bin/sh
date >> /tmp/test
echo '<table>'
echo '<tr><td> This is my custom script running </td></tr>'
echo '<tr><td>'
echo '<img src="http://vinras.com/images/linux-online-inc.jpg">'
echo '</td></tr>'
echo '</table>'
exit 0
```

CHAPTER 14: MANAGEMENT

The console server has a small number of Manage reports and tools that are available to both Administrators and Users to:

- Access and control authorized devices.
- View serial port logs and host logs for those devices.
- Use SSH or the Web Terminal to access serially attached consoles.
- Control of power devices (where authorized).

All other Management Console menu items are available to Administrators only.

14.1 DEVICE MANAGEMENT

NOTE: The Manage Devices UI has been significantly updated as of firmware version 3.12.

To display Managed Devices and their grouped serial, network and power connections:

- Select Manage > Devices or click the Manage Devices icon in the top right of the UI.
- admin-group users are presented with a list of all configured Managed Devices and their constituent connections. user-group users only see the Managed Devices where, for each Related Connection, they have been explicitly permitted access.

  The Status column displays the current most salient status for each Related Connection (for example, Active Users for serial connections, and power status for RPC outlet connections) with links to detailed status.

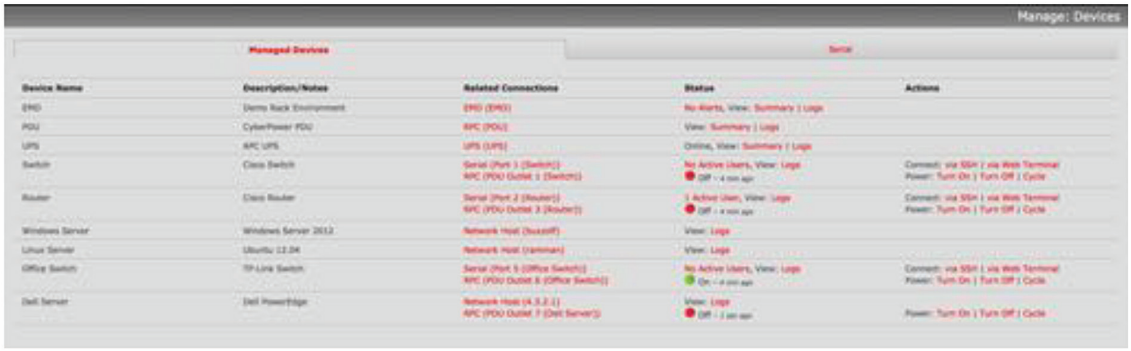


FIGURE 14-1.

- The links in the Actions column are used to control the Managed Device (for example, to connect to a console session or power cycle. Power actions are not performed until the action has been confirmed via pop-up message.)
- Alternatively, select the Serial tab for an ungrouped view of permitted serial port connections for the current user.
- An additional Signals column displays the current state of the serial pins.

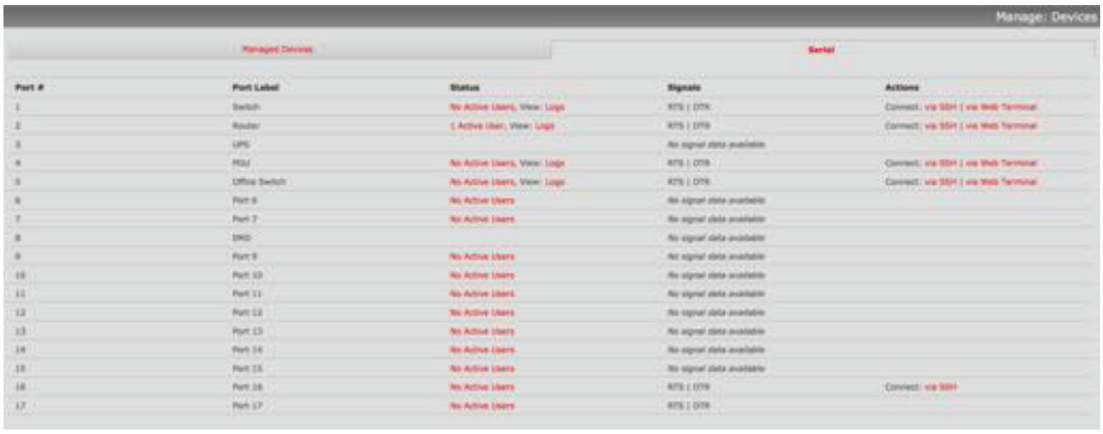


FIGURE 14-2.

NOTE: To use the Connect > via SSH links, your computer's operating system must recognize the ssh:// URI scheme and have a protocol handler configured (for example, an SSH client like SecureCRT).

14.2 PORT AND HOST LOGS

Administrators and Users can view and download logs of data transferred to and from connected devices.

- Select Manage > Port Logs.
- Select the serial Port # to be displayed.

To display Host logs:

- Select Manage > Host Logs.
- Select the Host to be displayed.

14.3 TERMINAL CONNECTION

There are two methods for accessing the console server command line and devices attached to the console server serial ports from a web browser.

The Web Terminal service uses AJAX to enable the browser to connect to the console server using HTTP or HTTPS, as a terminal without additional client installation on the user's PC. Browser access is available to users who are a member of the admin or users groups.

The SDT Connector service launches a pre-installed SDT Connector client on the user's PC to establish SSH access, then uses pre-installed client software on the client PC to connect to the console server.

14.3.1 WEB TERMINAL

- The AJAX based Web Terminal service may be used to access the console server command line or attached serial devices.

NOTE: Any communication using the Web Terminal service using HTTP is unencrypted and not secure. The Web Terminal connects to the command line or serial device using the same protocol that is being used to browse to the Black Box Management Console. If you browse using an https:// URL (this is the default), the Web Terminal connects using HTTPS.

To enable the Web Terminal service for the console server:

- Select System > Firewall.
- Check Enable Web Terminal.

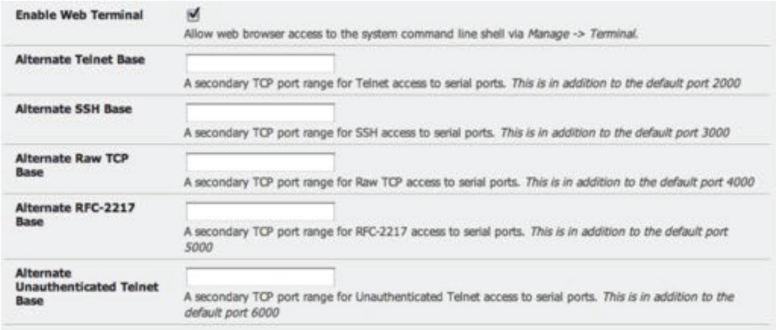


FIGURE 14-3.

- Click Apply.

Administrators can now communicate with the console server shell from their browser.

- Select Manage > Terminal to display the Web Terminal from which you can log in to the console server command line.

To enable the Web Terminal service for each serial port you want to access:

- Select Serial & Network > Serial Port.
- Click Edit.
- Ensure the serial port is in Console Server Mode.
- Check Web Terminal.
- Click Apply.

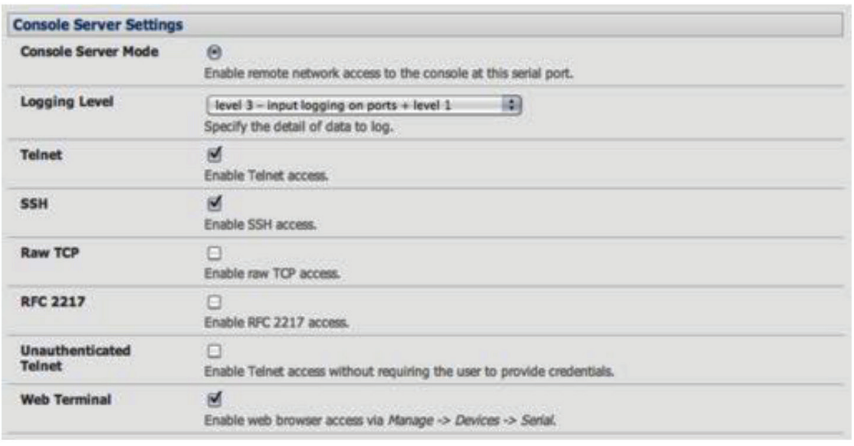


FIGURE 14-4.

Administrator and Users can communicate directly with serial port attached devices from their browser:

- Select Manage > Devices.
- Select the Serial tab.
- Under the Action column, click the Web Terminal icon to display the Web Terminal, connected directly to the attached serial device.

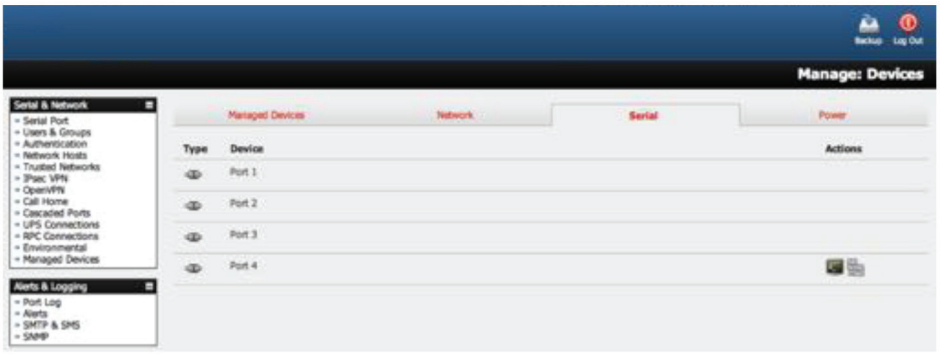


FIGURE 14-5.

NOTE: The Web Terminal feature was introduced in firmware v3.3. Earlier releases had an open source jcterm java terminal applet that could be downloaded into your browser to connect to the console server and attached serial port devices. jcterm had some JRE compatibility issues and is no longer supported.

14.3.2 SDT CONNECTOR ACCESS

Administrators and Users can communicate directly with the console server command line and with devices attached to the console server serial ports using SDT Connector and their local telnet client.

- Select Manage > Terminal.
- Click Connect to SDT Connector.

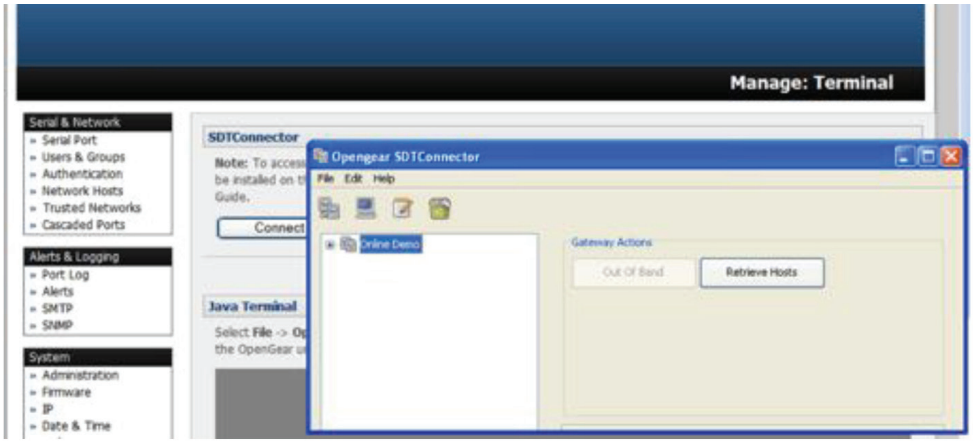


FIGURE 14-6.

This activates the SDT Connector client on the computer you are browsing and loads your local telnet client to connect to the command line or serial port using SSH.

NOTE: SDT Connector must be installed on the computer you are browsing from and the console server must be added as a gateway, as detailed in Chapter 7.

14.4 POWER MANAGEMENT

Administrators and Users can access and manage the connected power devices.

- Select Manage > Power.

This enables the user to Turn On, Turn Off, or Cycle the power on any power outlet on any PDU the user has been given access privileges to.

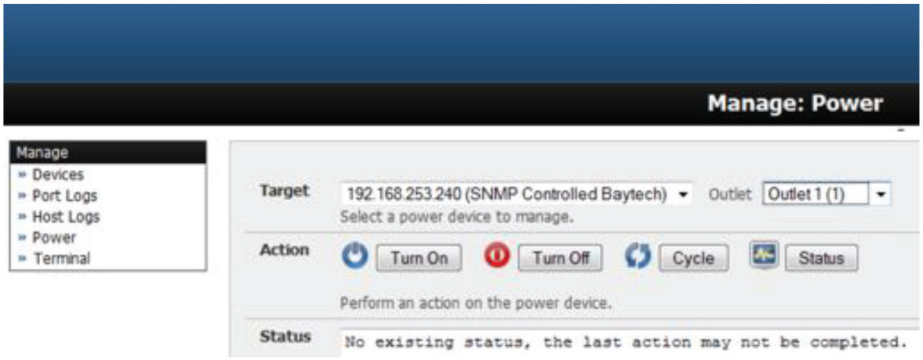


FIGURE 14-7.

See Chapter 9 for details.

CHAPTER 15: CONFIGURATION FROM THE COMMAND LINE

For those who prefer to configure their console server at the Linux command line level, rather than use a browser and the Management Console, this chapter describes using command line access and the config tool to manage the console server, configure the ports and so on.

This chapter walks through command line configuration to deliver the functions provided otherwise using the Management Console GUI.

For advanced and custom configurations and for details using other tools and commands, see Chapter 16.

When displaying a command, the convention used in the rest of this chapter is to use single quotes ('') for user-defined values (for example, descriptions and names). Element values without single quotes should be typed exactly as shown.

After the initial section on accessing the config command the menu items in this document follow the same structure as the menu items in the web GUI.

15.1 ACCESSING CONFIG FROM THE COMMAND LINE

The console server runs a standard Linux kernel and embeds a suite of open source applications. If you do not want to use a browser and the Management Console tools, you are free to configure the console server and to manage connected devices from the command line using standard Linux and Busybox commands and applications such as ifconfig, gettyd, stty, powerman, nut etc. Without care, these configurations may not withstand a power-cycle-reset or reconfigure.

Black Box provides a number of custom command line utilities and scripts to make it simple to configure the console server and ensure the changes are stored in the console server's flash memory etc.

In particular, the config utility allows manipulation of the system configuration from the command line. With config, a new configuration can be activated by running the relevant configurator, which performs the action necessary to make the configuration changes live.

To access config from the command line:

- Power up the console server and connect the “terminal” device.

  If you are connecting using the serial line, plug a serial cable between the console server’s local DB9 console port and terminal device. Configure the serial connection of the terminal device you are using to 115200 bps, 8 data bits, no parity and one stop bit.

  If you are connecting over the LAN, interconnect the Ethernet ports and direct your terminal emulator program to the IP address of the console server (192.168.0.1 by default).

- Log on to the console server by pressing Return a few times.

  The console server will request a username and password.

- Enter the username root and the password default.

  The command line prompt appears:

  #

NOTE: This chapter is not intended to teach you Linux. We assume you already have a certain level of understanding before you execute kernel-level Linux commands.

THE CONFIG TOOL

config [-ahv] [-d id] [-g id] [-p path] [-r configurator] [-s id=value] [-P id]

The config tool is designed to perform multiple actions from one command if need be, so if necessary, options can be chained together.

The config tool allows manipulation and querying of the system configuration from the command line. Using config, the new configuration can be activated by running the relevant configurator that performs the action necessary to make the configuration changes live.

The custom user configuration is saved in the /etc/config/config.xml file. This file is transparently accessed and edited when configuring the device using the Management Console browser GUI. Only the root user can configure from the shell.

By default, the config elements are separated by a . character (a full-stop or period). The root of the config tree is called <config>. To address a specific element, place a . between each node or branch. For example, to access and display the description of user1 type:

```
# config -g config.users.user1.description
```

The root node of the config tree is <config>. To display the entire config tree, type:

```
# config -g config
```

To display the help text for the config command, type:

```
# config -h
```

The config application resides in the /bin directory. The environmental variable called PATH contains a route to the /bin directory. This allows a user to simply type config at the command prompt instead of the full /bin/config path.

TABLE 15-1. CONFIG OPTIONS

OPTION	DESCRIPTION
-a --run-all	Run all registered configurators. This performs every configuration synchronization action pushing all changes to the live system.
-h --help	Display a brief usage message.
-v --verbose	Log extra debug information.
-d --del=id	Remove the given configuration element specified by a . separated identifier.
-g --get=id	Display the value of a configuration element
-p --path=file	Specify an alternate configuration file to use. The default file is located at /etc/config/config.xml.
-r --run=configurator	Run the specified registered configurator. Registered configurators are listed below.
-s --set=id=value	Change the value of configuration element specified by a . separated identifier.
-e --export=file	Save active configuration to file.
-i --import=file	Load configuration from file.
-t --test-import=file	Pretend to load configuration from file.
-S --separator=char	The pattern to separate fields with. The default is .
-P --password=id	Prompt user for a value. Hash the value, then save it in id.

The registered configurators are:

alerts
auth
cascade
console
dhcp
dialin
eventlog
hosts
ipaccess
ipconfig
nagios
power
serialconfig
services
slave
systemsettings
time
ups
users

There are three ways to delete a config element value. The simplest way is use the delete-node script detailed later in Chapter 16. You can also assign the config element to "" (null), or delete the entire config node using -d:

```
# /bin/config -d 'element name'
```

Passwords are saved as plaintext, except the user passwords and the system passwords, which are encrypted. As of firmware 3.16.6u1, however, password obfuscation is supported.

The config command does not verify whether the nodes edited/added by the user are valid. This means that any node may be added to the tree. If a user were to run the following command:

```
# /bin/config -s config.fruit.apple=sweet
```

the configurator will not complain, but this command is clearly useless. When the configurators are run (to turn the config.xml file into live config) they will simply ignore this <fruit> node. Administrators must make sure of the spelling when typing config commands. Incorrect spelling for a node will not be flagged.

Most configurations made to the XML file will be immediately active. To make sure that all configuration changes are active, especially when editing user passwords, run all the configurators:

```
# /bin/config -a
```

For information on backing up and restoring the configuration file see Chapter 16.

15.1.1 SERIAL PORT CONFIGURATION

The first set of configurations that needs to be made to any serial port are the RS-232 common settings. For example, you can set up serial port 5 to use the following properties:

Baud Rate	9600
Parity	None
Data Bits	8
Stop Bits	1
Label	Myport
Log level	0
Protocol	RS232
Flow control	None

To do this, use the following commands:

```
# config -s config.ports.port5.speed=9600
# config -s config.ports.port5.parity=None
# config -s config.ports.port5.charsize=8
# config -s config.ports.port5.stop=1
# config -s config.ports.port5.label=myport
# config -s config.ports.port5.loglevel=0
# config -s config.ports.port5.protocol=RS232
# config -s config.ports.port5.flowcontrol=None
```

The following command will synchronize the live system with the new configuration:

```
# config -r serialconfig
```

TABLE 15-2. SUPPORTED PROPERTIES

PROPERTY	SUPPORTED VALUES
baud rate	50. 75, 110, 124, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400
parity values	None, Odd, Even, Mark, Space
data bits	8, 7, 6, 5
stop bits	1, 1.5, 2
flow control	Hardware, Software, None

Additionally, before any port can function properly, the mode of the port needs to be set. Any port can be set to run in one of the five possible modes (see Chapter 5 for details):

- Console Server mode
- Device mode
- SDT mode
- Terminal server mode
- Serial bridge mode

All these modes are mutually exclusive.

CONSOLE SERVER MODE

The command to set the port in portmanager mode:

```
# config -s config.ports.port5.mode=portmanager
```

To set the following optional config elements for this mode:

Data accumulation period 1	00 ms
Escape character	% (default is ~)
l og level	2 (default is 0)
Shell power command menu	Enabled
RFC2217 access	Enabled
Limit port to 1 connection	Enabled
SSH access	Enabled
TCP access	Enabled
telnet access	Disabled
Unauthorized telnet access	Disabled

Run the following commands.

```
# config -s config.ports.port5.delay=100
# config -s config.ports.port5.escapechar=%
# config -s config.ports.port5.loglevel=2
# config -s config.ports.port5.powermenu=on
# config -s config.ports.port5.rfc2217=on
# config -s config.ports.port5.singleconn=on
```

```
# config -s config.ports.port5.ssh=on
# config -s config.ports.port5.tcp=on
# config -d config.ports.port5.telnet
# config -d config.ports.port5.unauthtel
```

DEVICE MODE

For a device mode port, set the port type to either ups, rpc, or enviro:

```
# config -s config.ports.port5.device.type=[ups | rpc | enviro]
```

For port 5 as a UPS port:

```
# config -s config.ports.port5.mode=reserved
```

For port 5 as an RPC port:

```
# config -s config.ports.port5.mode=powerman
```

For port 5 as an Environmental port:

```
# config -s config.ports.port5.mode=reserved
```

SDT MODE

To enable access over SSH to a host connected to serial port 5:

```
# config -s config.ports.port5.mode=sdt
# config -s config.ports.port5.sdt.ssh=on
```

To configure a username and password when accessing this port with Username = user1 and Password = secret:

```
# config -s config.ports.port#.sdt.username=user1
# config -s config.ports.port#.sdt.password=secret
```

TERMINAL SERVER MODE

Enable a TTY login for a local terminal attached to serial port 5:

```
# config -s config.ports.port5.mode=terminal
# config -s config.ports.port5.terminal=\
[vt220 | vt102 | vt100 | linux | ansi]
The default terminal is vt220.
```

SERIAL BRIDGE MODE

Create a network connection to a remote serial port via RFC-2217 on port 5:

```
# config -s config.ports.port5.mode=bridge
```

Optional configurations for the network address of RFC-2217 server of 192.168.3.3 and TCP port used by the RFC-2217 service = 2500:

```
# config -s config.ports.port5.bridge.address=192.168.3.3
# config -s config.ports.port5.bridge.port=2500
```

To enable RFC-2217 access:

```
# config -s config.ports.port5.bridge.rfc2217=on
```

To redirect the serial bridge over an SSH tunnel to the server:

```
#config -s config.ports.port5.bridge.ssh.enabled=on
```

SYSLOG SETTINGS

Additionally, the global system log settings can be set for any specific port, in any mode:

```
# config -s config.ports.port#.syslog.facility='facility'
# config -s config.ports.port#.syslog.priority='priority'
```

TABLE 15-3. GLOBAL SYSTEM SETTINGS

ARGUMENT	SUPPORTED VALUES
facility	Default, local 0–7, auth, authpriv, cron, daemon, ftp, kern, lpr, mail, news, user, uucp,
priority	Default, warning, notice, info, error, emergency, debug, critical, alert

15.1.2 ADDING AND REMOVING USERS

First, determine the total number of existing Users:

```
# config -g config.users.total
```

This command should display

```
config.users.total 1.
```

NOTE: If you see config.users.total, this means you have 0 Users configured.

Your new User will be the existing total plus 1. So if the previous command gave you 0 then you start with user number 1, if you already have 1 user your new user will be number 2, etc.

Assuming the previous command did return config.users.total 1., to add a user with Username=John, Password=secret and Description=mySecondUser issue the commands:

```
# config -s config.users.total=2
# config -s config.users.user2.username=John
# config -s config.users.user2.description=mySecondUser
# config -P config.users.user2.password
```

NOTE: The -P parameter will prompt the user for a password, and encrypt it. The value of any config element can be encrypted using the -P parameter, but only encrypted user passwords and system passwords are supported. If any other element value were to be encrypted, the value will become inaccessible and will have to be reset.

To add this user to specific groups (admin/users):

```
# config -s config.users.user2.groups.group1='groupname'
# config -s config.users.user2.groups.group2='groupname2'
# [etc...]
```

To give this user access to a specific port:

```
# config -s config.users.user2.port1=on
# config -s config.users.user2.port2=on
# config -s config.users.user2.port5=on
# [etc...]
```

To remove port access:

```
# config -s config.users.user2.port1=''
```

NOTE: The port1 value is left blank.

or simply:

```
# config -d config.users.user2.port1
```

The port number can be anything from 1 to 48, depending on the available ports on the specific console server.

For example, assume we have an RPC device connected to port 1 on the console server and the RPC is configured. To give this user access to RPC outlet number 3 on the RPC device, run the 2 commands below:

```
# config -s config.ports.port1.power.outlet3.users.user2=John
# config -s config.ports.port1.power.outlet3.users.total=2
```

The last command sets the total number of users with access to this outlet. If more users are given access, increment config.ports.port1.power.outlet3.users.total accordingly.

To give this user access to network host 5 (assuming the host is configured):

```
# config -s config.sdt.hosts.host5.users.user1=John
# config -s config.sdt.hosts.host5.users.total=1
```

The last command sets the total number of users having access to host.

To give another user called Peter access to the same host:

```
# config -s config.sdt.hosts.host5.users.user2=Peter
# config -s config.sdt.hosts.host5.users.total=2
```

The last command sets the total number of users having access to host.

To edit any of the user element values, use the same approach as when adding user elements. That is, use the -s parameter. If any of the config elements do not exist, they will automatically be created.

To delete the user called John, use the delete-node script:

```
# /etc/scripts/delete-node config.users.user2
```

The following command will synchronize the live system with the new configuration:

```
# config -r users
```

15.1.3 ADDING AND REMOVING USER GROUPS

The console server is configured with a few default user groups (only two of these groups are visible in the Management Console GUI). To find out how many groups are already present:

```
# config -g config.groups.total
```

Assume this value is six. Make sure to number any new groups you create from seven onwards.

To add a custom group to the configuration with Group name=Group7, Group description=MyGroup and Port access= 1,5 issue the commands:

```
# config -s config.groups.group7.name=Group7
```

```
# config -s config.groups.group7.description=MyGroup
# config -s config.groups.total=7
# config -s config.groups.group7.port1=on
# config -s config.groups.group7.port5=on
```

Assume we have an RPC device connected to port 1 on the console server, and the RPC is configured. To give this group access to RPC outlet number 3 on the RPC device, run the two commands below:

```
# config -s config.ports.port1.power.outlet3.groups.group1=Group7
# config -s config.ports.port1.power.outlet3.groups.total=1
```

The second command sets the total number of groups that have access to this outlet. If more groups are given access to this power outlet, then increment the config.ports.port1.power.outlet3.groups.total element accordingly.

To give this group access to network host 5:

```
# config -s config.sdt.hosts.host5.groups.group1=Group7
# config -s config.sdt.hosts.host5.groups.total=1
```

The second command sets the total number of groups with access to host.

To give another group called Group8 access to the same host:

```
# config -s config.sdt.hosts.host5.groups.group2=Group8
# config -s config.sdt.hosts.host5.groups.total=2
```

The second command sets the total number of groups with access to host.

To delete the group called Group7, use the following command:

```
# rmuser Group7
```

NOTE: The rmuser script is a generic script to remove any config element from config.xml correctly. Any dependencies or references to this group will not be affected. Only the group details are deleted. The administrator is responsible for going through config.xml and removing group dependencies and references manually, specifically if the group had access to a host or RPC device.

The following command will synchronize the live system with the new configuration:

```
# config -a
```

15.1.4 AUTHENTICATION

To change the type of authentication for the console server:

config -s config.auth.type='authtype'

'authtype' can be:

Local
LocalTACACS
TACACS
TACACSLocal
TACACSDownLocal
LocalRADIUS
RADIUS
RADIUSLocal

RADIUSDownLocal

LocalLDAP

LDAP

LDAPLocal

LDAPDownLocal

To configure TACACS authentication:

```
# config -s config.auth.tacacs.auth_server='comma-separated-list'
```

comma-separated-list is a list of remote authentiction and authorization servers.

```
# config -s config.auth.tacacs.acct_server='comma-separated-list'
# config -s config.auth.tacacs.password='password'
```

comma-separated-list is a list of remote accounting servers. If unset, the Authentication and Authorization Server Address will be used.

To configure RADIUS authentication:

```
# config -s config.auth.radius.auth_server='comma-separated-list'
# config -s config.auth.radius.acct_server='comma-separated-list'
# config -s config.auth.radius.password='password'
```

In the first command, comma-separated-list is a list of remote authentiction and authorization servers.

In the second command, comma-separated-list is a list of remote accounting servers. If unset, Authentication and Authorization Server Address will be used.

To configure LDAP authentication:

```
# config -s config.auth.ldap.server='comma separated list'
# config -s config.auth.ldap.basedn='name'
# config -s config.auth.ldap.binddn='name'
# config -s config.auth.radius.password='password'
```

In the first command, comma-separated-list is a list of remote servers,

In the second command, name is the distinguished name of the search base. For example: dc=my-company,dc=com.

In the third command, name is the distinguished name to bind to the server with. The default is to bind anonymously.

The following command will synchronize the live system with the new configuration:

```
# config -r auth
```

15.1.5 NETWORK HOSTS

To determine the total number of currently configured hosts:

```
# config -g config.sdt.hosts.total
```

Assume the value returned is 3. If you add another host, increment the total number of hosts from 3 to 4:

```
# config -s config.sdt.hosts.total=4
```

If the output is config.sdt.hosts.total, 0 hosts are configured.

ADD POWER DEVICE HOST

To add a UPS/RPC network host with the following details:

TABLE 15-4. UPS/RPC NETWORK HOST DETAILS

SETTING	VALUE
IP address or DNS name	192.168.2.5
Host name	remoteUPS
Description	UPSroom3
Type	UPS
Allowed services	ssh port 22 and https port 443
Log level for services	0

Issue the following commands:

```
# config -s config.sdt.hosts.host4.address=192.168.2.5
# config -s config.sdt.hosts.host4.name=remoteUPS
# config -s config.sdt.hosts.host4.description=UPSroom3
# config -s config.sdt.hosts.host4.device.type=ups
# config -s config.sdt.hosts.host4.tcpports.tcpport1=22
# config -s config.sdt.hosts.host4.tcpports.tcpport1.loglevel=0
# config -s config.sdt.hosts.host4.udpports.udpport2=443
# config -s config.sdt.hosts.host4.udpports.udpport2.loglevel=0
```

loglevel can have a value of 0 or 1.

The default services that should be configured are: 22/tcp (ssh), 23/tcp (telnet), 80/tcp (http), 443/tcp (https), 1494/tcp (ica), 3389/tcp (rdp), 5900/tcp (vnc).

ADD OTHER NETWORK HOST

To add any other type of network host with the following details:

TABLE 15-5. OTHER HOST DETAILS

SETTING	VALUE
IP address or DNS name	192.168.3.10
Host name	OfficePC
Description	MyPC
Allowed services	ssh port 22 and https port 443
Log level for services	1

Issue the commands below. If the Host is not a PDU or UPS power device or a server with IPMI power control then leave the device type blank:

```
# config -s config.sdt.hosts.host4.address=192.168.3.10
# config -s config.sdt.hosts.host4.description=MyPC
# config -s config.sdt.hosts.host4.name=OfficePC
# config -s config.sdt.hosts.host4.device.type=''
# config -s config.sdt.hosts.host4.tcpports.tcpport1=22
# config -s config.sdt.hosts.host4.tcpports.tcpport1.loglevel=1
# config -s config.sdt.hosts.host4.udpports.tcppport2=443
# config -s config.sdt.hosts.host4.udpports.tcpport2.loglevel=1
```

NOTE: Type should be left blank.

If you want to add the new host as a managed device, make sure to use the current total number of managed devices + 1, for the new device number.

To get the current number of managed devices:

```
# config -g config.devices.total
```

Assuming we already have one managed device, our new device will be device 2. Issue the following commands:

```
# config -s config. devices.device2.connections.connection1.name=192.168.3.10
# config -s config. devices.device2.connections.connection1.type=Host
# config -s config. devices.device2.name=OfficePC
# config -s config. devices.device2.description=MyPC
# config -s config.devices.total=2
```

The following command will synchronize the live system with the new configuration:

```
# config -r hosts
```

15.1.6 TRUSTED NETWORKS

You can further restrict remote access to serial ports based on the source IP address. To configure this via the command line you need to do the following:

Determine the total number of existing trusted network rules (if you have no existing rules) you can assume this is 0

```
# config -g config.portaccess.total
```

This command should display

```
config.portaccess.total 1
```

If you see config.portaccess.total you have 0 rules configured.

Your new rule will be the existing total plus 1. If the previous command gave you 0 start with rule number 1. If you already have 1 rule your new rule will be number 2. And so on.

Assuming you have a previous rule in place, if you want to restrict access to serial port 5 to computers from a single class C network (for example, 192.168.5.0) issue the following commands to add a trusted network:

```
# config -s config.portaccess.rule2.address=192.168.5.0
# config -s "config.portaccess.rule2.description=foo bar"
```

```
# config -s config.portaccess.rule2.netmask=255.255.255.0
# config -s config.portaccess.rule2.port5=on
# config -s config.portaccess.total=2
```

The following command will synchronize the live system with the new configuration:

```
# config -r serialconfig
```

15.1.7 CASCADED PORTS

To add a new slave device with the following settings:

TABLE 15-6. SLAVE DEVICE SETTINGS

SETTING	VALUE
IP address or DNS name	192.168.0.153
Description	Office 42
Label	LES1716A-R2
Number of ports	16

Issue the following commands:

```
# config -s config.cascade.slaves.slave1.address=192.168.0.153
# config -s "config.cascade.slaves.slave1.description=Office 42"
# config -s config.cascade.slaves.slave1.label=cm7116-5
# config -s config.cascade.slaves.slave1.ports=16
```

The total number of slaves must also be incremented. If this is the first slave being added, type:

```
# config -s config.cascade.slaves.total=1
```

Increment this value when adding more slaves.

NOTE: If a slave is added using the CLI, the master SSH public key will need to be manually copied to every slave device before cascaded ports will work (see Chapter 5).

The following command will synchronize the live system with the new configuration:

```
# config -r cascade
```

15.1.8 UPS CONNECTIONS

MANAGED UPSES

Before adding a managed UPS, make sure that at least 1 port has been configured to run in “device mode,” and that the device is set “ups.”

To add a managed UPS with the following values:

TABLE 15-7. MANAGED UPSES

SETTING	VALUE
Connected via	Port 1
UPS name	My UPS
Description	Room 5 UPS
Username to connect to UPS	user2
Password to connect to UPS	A-secret-for-2.
shutdown order	2 (0 shuts down first)
Driver	genericups
Driver option	option
Driver argument	argument
Logging	Enabled
Log interval	2 minutes
Run script when power is critical	Enabled

Run the following commands:

```
# config -s config.ups.monitors.monitor1.port=/dev/port01
# config -s "config.ups.monitors.monitor1.name=My UPS"
# config -s "config.ups.monitors.monitor1.description=Room 5 UPS"
# config -s config.ups.monitors.monitor1.username=user2
# config -s config.ups.monitors.monitor1.password=A-secret-for-2.
# config -s config.ups.monitors.monitor1.sdorder=2
# config -s config.ups.monitors.monitor1.driver=genericups
# config -s \
config.ups.monitors.monitor1.options.option1.opt=option
# config -s \
config.ups.monitors.monitor1.options.option1.arg=argument
# config -s config.ups.monitors.monitor1.options.total=1
# config -s config.ups.monitors.monitor1.log.enabled=on
# config -s config.ups.monitors.monitor1.log.interval=2
# config -s config.ups.monitors.monitor1.script.enabled=on
```

With regards the first command above, if the port number is higher than 9 (eg port 13) enter the command as follows:

```
# config -s config.ups.monitors.monitor1.port=/dev/port13
```

Also, make sure to increment the total monitors:

```
# config -s config.ups.monitors.total=1
```

Assuming there are already 2 managed devices configured, the 5 commands below will add the UPS to Managed Devices.

```
# config -s \
"config.devices.device3.connections.connection1.name=My UPS"
# config -s \
"config.devices.device3.connections.connection1.type=UPS Unit"
# config -s "config.devices.device3.name=My UPS"
# config -s "config.devices.device3.description=Room 5 UPS"
# config -s config.devices.total=3
```

To delete this managed UPS:

```
# config -d config.ups.monitors.monitor1
```

NOTE: Decrement monitors.total when deleting a managed UPS.

REMOTE UPSES

To add a remote UPS with the following details (assuming this is our first remote UPS):

TABLE 15-8. REMOTE UPSES

SETTING	VALUE
UPS name	oldUPS
Description	Room 2 UPS
Address	192.168.50.50
Log status	Disabled
Log rate	240 seconds
Run shutdown script	Enabled

```
# config -s config.ups.remotes.remote1.name=oldUPS
# config -s "config.ups.remotes.remote1.description=Room 2 UPS"
# config -s config.ups.remotes.remote1.address=192.168.50.50
# config -d config.ups.remotes.remote1.log.enabled
# config -s config.ups.remotes.remote1.log.interval=240
# config -s config.ups.remotes.remote1.script.enabled=on
# config -s config.ups.remotes.total=1
```

The following command will synchronize the live system with the new configuration:

```
# config -a
```

15.1.9 RPC CONNECTIONS

You can add an RPC connection from the command line but it is not recommended because of dependency issues.

Before adding an RPC, the Management Console GUI code makes sure that at least 1 port has been configured to run in device mode, and that the device is set to rpc. The CLI-based approach does not do this.

To add an RPC with the following values:

TABLE 15-9. RPC CONNECTIONS

SETTING	VALUE
RPC type	APC 7900
Connected via	Port 2
UPS name	MyRPC
Description	Room 5 RPC
Login name for device	rpclogin
Login password for device	A-secret-for-2.
SNMP community	v1 or v2c
Logging	option
Driver argument	argument
Logging	enabled
Log interval	600 seconds
Number of power outlets	4

Run the following commands:

```
# config -s config.ports.port2.power.type=APC 7900
# config -s config.ports.port2.power.name=MyRPC
# config -s "config.ports.port2.power.description=Room 5 RPC"
# config -s config.ports.port2.power.username=rpclogin
# config -s config.ports.port2.power.password=A-secret-for-2.
# config -s config.ports.port2.power.snmp.community=v1
# config -s config.ports.port2.power.log.enabled=on
# config -s config.ports.port2.power.log.interval=600
# config -s config.ports.port2.power.outlets=4
```

The following five commands are used by the Management Console to add the RPC to Managed Devices:

```
# config -s \
config.devices.device3.connections.connection1.name=myRPC
# config -s \
"config.devices.device3.connections.connection1.type=RPC Unit"
# config -s config.devices.device3.name=myRPC
```

```
# config -s "config.devices.device3.description=Room 5 RPC"
# config -s config.devices.total=3
```

The following command will synchronize the live system with the new configuration:

```
# config -a
```

15.1.10 MANAGED DEVICES

To add a managed device: (see Chapter 9 for more information):

```
# config -s "config.devices.device8.name=8"
# config -s "config.devices.device8.description=the-8th-device"
# config -s \
"config.devices.device8.connections.connection1.name=8"
# config -s \
config.devices.device8.connections.connection1.type=type
# config -s config.devices.total=8
```

type can be serial, Host, UPS, or RPC.

To delete the above managed device:

```
# config -d config.devices.device8
```

NOTE: The config.devices.total total must also be decremented when deleting a managed device.

The following command will synchronize the live system with the new configuration:

```
# config -a
```

15.1.11 PORT LOG

To configure serial/network port logging:

```
# config -s config.eventlog.server.address=remote-server-ip
# config -s config.eventlog.server.logfacility=facility
# config -s config.eventlog.server.logpriority=priority
```

facility and priority can take a specific range of values:

TABLE 15-10. PORT LOG

VARIABLE	ALLOWED VALUES
facility	Daemon, Local 0–7, Authenticaiton, Kernel, User, Syslog, Mail, News, UUCP
priority	Info, Alert, Critical, Debug, Emergency, Error, Notice, Warning

Assume the remote log server needs a username name1 and password A-secret-for-2.

```
# config -s config.eventlog.server.username=name1
# config -s config.eventlog.server.password=A-secret-for-2.
```

To set the remote path as /Black Box/logs to save logged data:

```
# config -s config.eventlog.server.path=/Black Box/logs
# config -s config.eventlog.server.type=[none|syslog|nfs|cifs|usb]
```

If the server type is set to usb, none of the other values need to be set. The mount point for storing on a remote USB device is /var/run/portmanager/logdir.

The following command will synchronize the live system with the new configuration:

```
# config -a
```

15.1.12 ALERTS

You can add an email, SNMP or NAGIOS alert by following the steps below.

THE GENERAL SETTING FOR ALL ALERTS

Assume this is our second alert, and we want to send email alerts to john@Black Box.com and sms alerts to peter@Black Box.com:

```
# config -s config.alerts.alert2.description=MySecondAlert
# config -s config.alerts.alert2.email=john@Black Box.com
# config -s config.alerts.alert2.email2=peter@Black Box.com
```

To use NAGIOS to notify of this alert

```
# config -s config.alerts.alert2.nsca.enabled=on
```

To use SNMP to notify of this alert

```
# config -s config.alerts.alert2.snmp.enabled=on
```

To increment the total alerts:

```
# config -s config.alerts.total=2
```

Below are the specific settings depending on the type of alert required.

CONNECTION ALERT

To trigger an alert when a user connects to serial port 5 or network host 3:

```
# config -s config.alerts.alert2.host3=hostname
# config -s config.alerts.alert2.port5=on
# config -s config.alerts.alert2.sensor=temp
# config -s config.alerts.alert2.signal=DSR
# config -s config.alerts.alert2.type=login
```

SIGNAL ALERT

To trigger an alert when a signal changes state on port 1:

```
# config -s config.alerts.alert2.port1=on
# config -s config.alerts.alert2.sensor=temp
# config -s config.alerts.alert2.signal=[DSR | DCD | CTS]
# config -s config.alerts.alert2.type=signal
```

PATTERN MATCH ALERT

To trigger an alert if the regular expression .*0.0% id is found in serial port 10's character stream.

```
# config -s "config.alerts.alert2.pattern=.*0.0% id"
# config -s config.alerts.alert2.port10=on
# config -s config.alerts.alert2.sensor=temp
# config -s config.alerts.alert2.signal=DSR
# config -s config.alerts.alert2.type=pattern
```

UPS POWER STATUS ALERT

To trigger an alert when myUPS (on localhost) or thatUPS (on remote host 192.168.0.50) power status changes between on line, on battery and low battery.

```
# config -s config.alerts.alert2.sensor=temp
# config -s config.alerts.alert2.signal=DSR
# config -s config.alerts.alert2.type=ups
# config -s config.alerts.alert2.ups1=myUPS@localhost
# config -s config.alerts.alert2.ups2=thatUPS@192.168.0.50
```

ALARM SENSOR ALERT

The commands below set an alert for doorAlarm and windowAlarm, two alarms connected to an environmental sensor called SensorInRoom3. Both alarms are disabled on Mondays from 08:15 to 14:30.

```
# config -s \
config.alerts.alert2.alarm1=SensorInRoom3.alarm1 (doorAlarm)
# config -s \
config.alerts.alert2.alarm1=SensorInRoom3.alarm2 (windowAlarm)
# config -s config.alerts.alert2.alarmrange.mon.from.hour=8
# config -s config.alerts.alert2.alarmrange.mon.from.min=15
# config -s config.alerts.alert2.alarmrange.mon.until.hour=14
# config -s config.alerts.alert2.alarmrange.mon.until.min=30
# config -s config.alerts.alert2.description='description'
# config -s config.alerts.alert2.sensor=temp
# config -s config.alerts.alert2.signal=DSR
# config -s config.alerts.alert2.type=alarm
```

To enable an alarm for the entire day:

```
# config -s config.alerts.alert2.alarmrange.mon.from.hour=0
# config -s config.alerts.alert2.alarmrange.mon.from.min=0
# config -s config.alerts.alert2.alarmrange.mon.until.hour=0
# config -s config.alerts.alert2.alarmrange.mon.until.min=0
```

The following command will synchronize the live system with the new configuration:

```
# config -r alerts
```

15.1.13 SMTP AND SMS

To set-up an SMTP mail or SMS server with the following details:

TABLE 15-11. SMTP OR SMS SETTINGS

SMTP OR SMS SERVER SETTING	VALUE
Outgoing server address	mail.Black Box.com
Secure connection type	SSL
Sender	john@Black Box.com
Server username	john
Server password	A-little-secret-for-2.
Subject line	SMTP alerts.

Run the following commands:

```
# config -s config.system.smtp.server=mail.Black Box.com
# config -s config.system.smtp.encryption=SSL
# config -s config.system.smtp.sender=John@Black Box.com
# config -s config.system.smtp.username=john
# config -s config.system.smtp.password=A-little-secret-for-2.
# config -s config.system.smtp.subject=SMTP alerts
```

To set-up an SMTP SMS server with the same details as above:

```
# config -s config.system.smtp.server2=mail.Black Box.com
# config -s config.system.smtp.encryption2=SSL
# config -s config.system.smtp.sender2=john@Black Box.com
# config -s config.system.smtp.username2=john
# config -s config.system.smtp.password2=A-little-secret-for-2.
# config -s config.system.smtp.subject2=SMTP alerts
```

In both setups, the value for encryption can be SSL, TLS or None.

The following command will synchronize the live system with the new configuration:

```
# config -a
```

15.1.14 SNMP

To set up the SNMP agent on the device:

```
# config -s config.system.snmp.protocol=[UDP | TCP]
# config -s config.system.snmp.trapport=port-number
# config -s config.system.snmp.address=NMS-IP-network-address
# config -s config.system.snmp.community=community-name
# config -s config.system.snmp.engineid=ID
# config -s config.system.snmp.username=username
# config -s config.system.snmp.password=password
# config -s config.system.snmp.version=[1 | 2c | 3]
```

The default port number is 162.

The community value can only be set on v1 and v2c.

The engineid, username, and password values can only be set on v3.

The following command will synchronize the live system with the new configuration:

```
# config -r auth
```

15.1.15 ADMINISTRATION

To change the administration settings to:

TABLE 15-12. ADMINISTRATION SETTINGS

SYSTEM SETTING	VALUE
System name	og.example.com
System password (root account password)	A-simple-little-secret-for-2.
Description	Device in office 2

Run the following commands:

```
# config -s config.system.name=og.example.com:
# config -P config.users.user1.password
# config -s "config.system.location="Device in office 2"
```

The second command has an interactive aspect. The -P parameter will prompt the user for a password. Enter the desired string and press Return: config will accept and encrypt the string.

NOTE: Any config element value can be encrypted using the -P parameter. Only encrypted user passwords and system passwords are supported, however. If any other element value is encrypted, the value becomes inaccessible and will have to be re-set.

An alternative to the second command above is:

```
# /etc/scripts/user-mod -P root
```

The following command will synchronize the live system with the new configuration:

```
# config -r auth
```

15.1.16 IP SETTINGS

To configure the primary network interface with the following static settings:

TABLE 15-13. IP SETTINGS

NETWORK INTERFACE SETTING	VALUE
IP address	192.168.0.23
Netmask	255.255.255.0
Default gateway	192.168.0.1
DNS server 1	192.168.0.1
DNS server 2	192.168.0.2

```
# config -s config.interfaces.wan.address=192.168.0.23
# config -s config.interfaces.wan.netmask=255.255.255.0
# config -s config.interfaces.wan.gateway=192.168.0.1
# config -s config.interfaces.wan.dns1=192.168.0.1
# config -s config.interfaces.wan.dns2=192.168.0.2
# config -s config.interfaces.wan.mode=static
# config -s config.interfaces.wan.media=<value>
```

In the last command, the available options for <value> are: Auto, 100baseTx-FD, 100baseTx-HD, 10baseT-HD, and 10baseT-FD.

To configure the management LAN interface, use the same commands as above but replace config.interfaces.wan with config.interfaces.lan.

To enable bridging between all interfaces:

```
# config -s config.system.bridge.enabled=on
```

To enable IPv6 for all interfaces:

```
# config -s config.system.ipv6.enabled=on
```

To enable the management LAN interface run the following command:

```
# config -d config.interfaces.lan.disabled
# config -r ipconfig
```

NOTE: Not all devices have a management LAN interface.

To configure a failover device in case of an outage:

```
# config -s config.interfaces.wan.failover.address1=ip-address
# config -s config.interfaces.wan.failover.address2=ip-address
# config -s config.interfaces.wan.failover.interface=<interface>
```

In the last command, the available options for <interface> are: eth1, console, modem

Network interfaces can also be configured automatically:

```
# config -s config.interfaces.wan.mode=dhcp
# config -s config.interfaces.lan.mode=dhcp
```

Either of the following commands will synchronize the live system with the new configuration:

```
# /bin/config –-run=ipconfig
# config -r ipconfig
```

15.1.17 DATE AND TIME SETTINGS

To enable NTP using a server at pool.ntp.org, issue the following commands:

```
# config -s config.ntp.enabled=on
# config -s config.ntp.server=pool.ntp.org
```

Alternatively, you can manually change the clock settings.

To change running system time:

```
# date MMDDhhmm[CC]YY.ss
# /bin/hwclock --systohc
```

The first command sets a new system time.

NOTE: The date command uses a US-style order with month (MM) listed before day (DD). Also, although the thousands and hundreds column in the Gregorian Year are theoretically optional, it is strongly recommended that these values be set explicitly.

The second command saves this new system time to the hardware clock.

Alternatively, to first change the hardware clock and then set the system time to the newly set hardware time :

```
# /bin/hwclock --set --date=MMDDhhmm[CC]YY.ss
# /bin/hwclock --hctosys
```

To change the timezone:

```
# config -s config.system.timezone=US/Eastern
```

The following command will synchronize the live system with the new configuration:

```
# config -r time
```

15.1.18 DIAL-IN SETTINGS

To enable dial-in access on the DB9 serial port from the command line with the following attributes:

TABLE 15-14. DIAL-IN SETTINGS

SETTING	VALUE
Local IP address	172.24.1.1
Remote IP address	172.24.1.2
Authentication type	MSCHAPv2
Serial port baud rate	115200
Serial port flow control	Hardware
Custom modem initialization	ATQ0V1H0
Callback phone number	0800223665
User to dial as	user1
Password for user	A-little-secret-for-2.

Run the following commands:

```
# config -s config.console.ppp.localip=172.24.1.1
# config -s config.console.ppp.remoteip=172.24.1.2
# config -s config.console.ppp.auth=MSCHAPv2
# config -s config.console.speed=115200
# config -s config.console.flow=Hardware
# config -s config.console.initstring=ATQ0V1H0
# config -s config.console.ppp.enabled=on
# config -s config.console.ppp.callback.enabled=on
# config -s config.console.ppp.callback.phone1=0800223665
# config -s config.console.ppp.username=user1
# config -s config.console.ppp.password=A-little-secret-for-2.
```

To make the dialed connection the default route:

```
# config -s config.console.ppp.defaultroute=on
```

Supported values for settings that are not fixed or user-created are as follows:

TABLE 15-15. SUPPORTED VALUES

SETTING	SUPPORTED VALUE
Authentication type	None, PAP, CHAP, and MSCHAPv2.
Serial port baud rate	9600, 19200, 38400, 57600, 115200, and 230400
Parity values	None, Odd, Even, Mark, and Space
Data bits values	5, 6, 7, and 8
Stop-bit values	1, 1.5, and 2
Serial port flow control	Hardware, Software, and None

If you do not wish to use out-of-band dial-in access the procedure for enabling start-up messages on the console port is documented in chapter 14.3.2.

The following command will synchronize the live system with the new configuration:

```
# config –a
```

15.1.19 DHCP SERVER

To enable the DHCP server on the console management LAN with the following settings:

TABLE 15-16. DHCP SERVER SETTINGS

DHCP SERVER SETTING	VALUE
Default lease time	200000 seconds.
Maximum lease time	300000 seconds
DNS server 1	192.168.2.3
DNS server 2	192.168.2.4
Domain name	company.com
Default gateway	192.168.0.1
IP pool 1 start address	192.168.0.20
IP pool 1 end address	192.168.0.100
Reserved IP address	192.168.0.50
MAC to reserve IP for	00:1e:67:82:72:d9
Name to identify this host	John-PC

Run the following commands:

```
# config -s config.interfaces.lan.dhcpd.enabled=on
# config -s config.interfaces.lan.dhcpd.defaultlease=200000
# config -s config.interfaces.lan.dhcpd.maxlease=300000
# config -s config.interfaces.lan.dhcpd.dns1=192.168.2.3
# config -s config.interfaces.lan.dhcpd.dns2=192.168.2.4
# config -s config.interfaces.lan.dhcpd.domain=company.com
# config -s config.interfaces.lan.dhcpd.gateway=192.168.0.1
# config -s \
config.interfaces.lan.dhcpd.pools.pool1.start=192.168.0.20
# config -s \
config.interfaces.lan.dhcpd.pools.pool1.end=192.168.0.100
# config -s config.interfaces.lan.dhcpd.pools.total=1
# config -s \
config.interfaces.lan.dhcpd.staticips.staticip1.ip=192.168.0.50
# config -s \
config.interfaces.lan.dhcpd.staticips.staticip1.mac=00:1e:67:82:72:d9
# config -s \
config.interfaces.lan.dhcpd.staticips.staticip1.host=John-PC
# config -s config.interfaces.lan.dhcpd.staticips.total=1
```

The following command will synchronize the live system with the new configuration:

```
# config -r auth
```

15.1.20 SERVICES

You can manually enable or disable network servers from the command line. For example, if you wanted to guarantee the following server configuration:

TABLE 15-17. SERVER CONFIGURATION

SERVER	STATE
HTTP server	enabled
HTTPS server	disabled
Telnet server	disabled
SSH server	enabled
SNMP server	disabled
Respond to ICMP echo requests (Ping replies)	disabled
TFTP server	enabled

Run the following commands:

```
# config -s config.services.http.enabled=on
# config -d config.services.https.enabled
# config -d config.services.telnet.enabled
# config -s config.services.ssh.enabled=on
# config -d config.services.snmp.enabled
# config -d config.services.pingreply.enabled
# config -s config.services.tftp.enabled=on
```

These services run on default port numbers as follows:

TABLE 15-18. DEFAULT PORT NUMBERS

SERVICE	DEFAULT PORT NUMBER
Telnet	2000
SSH	3000
TCP	4000
RFC2217	5000
unauthtel (Unauthorized Telnet)	6000

To set secondary port ranges for any service the following syntax applies:

```
# config -s config.services.<service>.portbase=<number>
```

For example: to set all these services to run on a port number that is ten higher than their default, run the following commands:

```
# config -s config.services.telnet.portbase=2010
# config -s config.services.ssh.portbase=3010
# config -s config.services.tcp.portbase=4010
# config -s config.services.rfc2217.portbase=5010
# config -s config.services.unauthtel.portbase=6010
```

The following command will synchronize the live system with the new configuration:

```
# config -r auth
```

15.1.21 NAGIOS

To configure NAGIOS with the following settings:

TABLE 15-19. NAGIOS SETTINGS

SETTING	VALUE	NOTES
NAGIOS host name	LES1716A-R2	Name of this system
NAGIOS host address	192.168.0.1	Address of this system
NAGIOS server address	192.168.0.10	Address of upstream NAGIOS server
Enable SDT for NAGIOS ext	Enabled	—
SDT gateway address	192.168.0.1	Defaults to host address
Prefer NRPE over NSCA	Disabled	Defaults to disabled

Run the following commands:

```
# config -s config.system.nagios.enabled=on
# config -s config.system.nagios.name=cm7116
# config -s config.system.nagios.address=192.168.0.1
# config -s config.system.nagios.server.address=192.168.0.10
# config -s config.system.nagios.sdt.disabled=on
# config -s config.system.nagios.sdt.address=192.168.0.1
# config -s config.system.nagios.nrpe.prefer=''
```

The fifth command disables SDT for Nagios extensions.

To configure NRPE with following settings:

TABLE 15-20. NRPE SETTINGS

SETTING	VALUE	NOTES
NRPE port	5600	Port to listen on for nrpe. Defaults to 5666.
NRPE user	user1	User to run as. Defaults to nrpe.
NRPE group	group1	Group to run as. Defaults to nobody.
Allow command arguments	Enabled	—

Run the following commands:

```
# config -s config.system.nagios.nrpe.enabled=on
# config -s config.system.nagios.nrpe.port=5600
# config -s config.system.nagios.user=user1
# config -s config.system.nagios.nrpe.group=group1
# config -s config.system.nagios.nrpe.cmdargs=on
```

To configure NSCA with the following settings:

TABLE 15-21. NCSA SETTINGS

SETTING	VALUE	NOTES
NSCA encryption	BLOWFISH	can be None, XOR, DES, TRPLEDES, CAST-256, BLOWFISH, TWOFISH, RIJNDAEL-256, SERPENT, GOST
NSCA password	secret	—
NSCA check-in interval	2 minutes	—
NSCA port	5650	Defaults to 5667
User to run as	user1	Defaults to nsca
Group to run as	group1	Defaults to nobody

Run the following commands

```
# config -s config.system.nagios.nsca.enabled=on
# config -s config.system.nagios.nsca.encryption=BLOWFISH
# config -s config.system.nagios.nsca.secret=secret
# config -s config.system.nagios.nsca.interval=2
# config -s config.system.nagios.nsca.port=5650
# config -s config.system.nagios.nsca.user=user1
# config -s config.system.nagios.nsca.group=group1
```

The following command will synchronize the live system with the new configuration:

```
# config -r auth
```

CHAPTER 16: ADVANCED CONFIGURATION

Black Box console servers run the embedded Linux operating system. Administrator class users can configure the console server and monitor and manage attached serial console and host devices from the command line using Linux commands and the config utility (as described in Chapter 15).

The Linux kernel in the console server also supports GNU bash shell scripts, enabling the Administrator to run custom scripts. This chapter presents a number of useful scripts and scripting tools including

- delete-node, which is a general script for deleting users, groups, hosts, UPS's etc.
- ping-detect, which will run specified commands when a specific host stops responding to ping requests.

This chapter then details how to perform advanced and custom management tasks using Black Box commands, Linux commands and the open source tools embedded in the console server:

- portmanager serial port management.
- raw data access to the ports and modems.
- iptables modifications and updating IP filtering rules.
- retrieving status information using SNMP and modifying SNMP with net-snmpd.
- public key authenticated SSH communications.
- SSL, configuring HTTPS and issuing certificates.
- using pmpower for NUT and PowerMan power device management.
- using IPMItools.
- CDK custom development kit.
- sms server tools.
- disabling multicasting.

16.1 CUSTOM SCRIPTING

The console server supports GNU bash shell commands (see Appendix A), enabling the Administrator to run custom scripts.

16.1.1 CUSTOM SCRIPT TO RUN WHEN BOOTING

The /etc/config/rc.local script runs whenever the system boots. By default, this script file is empty. You can add any commands to this file if you want them to be run at boot time. For example, if you want to display hello world, add the following to rc.local:

```
#!/bin/sh
echo "Hello World!"
```

If this script has been copied from a Windows machine, you may need to run the following command on the script before bash can run it successfully:

```
# dos2unix /etc/config/rc.local
```

Another scenario would be to call another custom script from the /etc/config/rc.local file, ensuring that your custom script will run whenever the system is booted.

16.1.2 RUNNING CUSTOM SCRIPTS WHEN ALERTS ARE TRIGGERED

Whenever an alert gets triggered, specific scripts get called. These scripts all reside in /etc/scripts/. Below is a list of the default scripts that get run for each applicable alert.

- For a connection alert (when a user connects or disconnects from a port or network host):

  /etc/scripts/portmanager-user-alert (for port connections).

  /etc/scripts/sdt-user-alert (for host connections).

- For a signal alert (when a signal on a port changes state):

  /etc/scripts/portmanager-signal-alert

- For a pattern match alert (when a specific regular expression is found in the serial ports character stream):

  /etc/scripts/portmanager-pattern-alert

- For a UPS status alert (when the UPS power status changes between on line, on battery, and low battery):

  /etc/scripts/ups-status-alert

- For an environmental, power and alarm sensor alerts (temperature, humidity, power load and battery charge alerts):

/etc/scripts/environmental-alert

- For an interface failover alert:

  /etc/scripts/interface-failover-alert

All these scripts do a check to see whether you have created a custom script to run instead. The code that does this check is shown below (an extract from the file /etc/scripts/portmanager-pattern-alert):

```
# If there's a user-configured script, run it instead
scripts[0]="/etc/config/scripts/pattern-alert.${ALERT_PORTNAME}"
scripts[1]="/etc/config/scripts/portmanager-pattern-alert"
for (( i=0 ; i < ${#scripts[@]} ; i++ )); do
        if [ -f "${scripts[$i]}" ]; then
        exec /bin/sh "${scripts[$i]}"
        fi
done
```

This code shows that there are two alternative scripts that can be run instead of the default one. This code first checks whether a file — /etc/config/scripts/pattern-alert.${ALERT_PORTNAME} — exists.

NOTE: The variable ${ALERT_PORTNAME} must be replaced with port01 or port13 or whichever port the alert should run for.

If this file cannot be found, the script checks whether the file /etc/config/scripts/portmanager-pattern-alert exists.

If either of these files exists, the script calls the exec command on the first file that it finds and runs that custom file/script instead.

As an example, you can copy the /etc/scripts/portmanager-pattern-alert script file to /etc/config/scripts/portmanager-pattern-alert:

```
# cd /
# mkdir /etc/config/scripts
```

NOTE: This command assumes the directory created does not already exist.

```
# cp /etc/scripts/portmanager-pattern-alert \
/etc/config/scripts/portmanager-pattern-alert
```

The next step is to edit the new script file.

- Open the file /etc/config/scripts/portmanager-pattern-alert using vi (or other text editor).
- Remove the lines that check for a custom script (the code from above).

This will prevent the new custom script from repeatedly calling itself.

After these lines have been removed, edit the file, or add any additional scripting to the file.

16.1.3 EXAMPLE SCRIPT: POWER CYCLING ON PATTERN MATCH

If, for example, we had an RPC (PDU) connected to port 1 on a console server and also have some telecommunications device connected to port 2 that is powered by the RPC outlet 3. Now assume the telecom device transmits a character stream Emergency out on its serial console port every time that it encounters some specific error, and the only way to fix this error is to power cycle the telecom device.

The first step is to setup a pattern-match alert on port 2 to check for the pattern Emergency.

Next, we need to create a custom script to deal with this alert:

```
# cd /
# mkdir /etc/config/scripts
/* if the directory does not already exist */
# cp /etc/scripts/portmanager-pattern-alert \
/etc/config/scripts/portmanager-pattern-alert
```

NOTE: Make sure to remove the if statement (which checks for a custom script) from the new script, to prevent an infinite loop.

The pmpower utility is used to send power commands to RPC device in order to power cycle our telecom device:

```
# pmpower -l port01 -o 3 cycle
```

The RPC is on serial port 1. The telecom device is powered by RPC outlet 3.

We can now append this command to our custom script. This will guarantee that our telecom device will be power cycled every time the console reads the Emergency character stream on port 2.

16.1.4 EXAMPLE SCRIPT: MULTIPLE E-MAIL NOTIFICATIONS ON EACH ALERT

If you desire to send more than one email when an alert triggers, you have to create a replacement script using the method described above and add the appropriate lines to your new script.

Currently, there is a script /etc/scripts/alert-email that runs from within all the alert scripts (for example portmanager-user-alert or environmental-alert). The alert-email script is responsible for sending the email. The line which invokes the email script looks as follows:

```
/bin/sh /etc/scripts/alert-email $suffix &
```

If you wish to send another email to a single address or the same email to many recipients, edit the custom script appropriately. You can follow the examples in any of the seven alert scripts listed above. Consider the portmanager-user-alert script. If you need to send the same alert email to more than one email address, find the lines in the script responsible for invoking the alert-email script, then add the following lines below the existing lines:

```
export TOADDR="emailaddress@domain.com"
/bin/sh /etc/scripts/alert-email $suffix &
```

These two lines assign a new email address to TOADDR and invoke the alert-email script in the background.

16.1.5 DELETING CONFIGURATION VALUES FROM THE CLI

The delete-node script is provided to help with deleting nodes from the command line. The delete-node script takes one argument, the node name you want to delete (for example config.users.user1 or config.sdt.hosts.host1).

delete-node is a general script for deleting any node you desire (users, groups, hosts, UPS's, etc.) from the command line. The script deletes the specified node and shuffles the remainder of the node values.

For example, if we have five users configured and we use the script to delete user 3, then user 4 will become user 3, and user 5 will become user 4.

This creates an obvious complication as this script does not check for any other dependencies that the node being deleted may have had. So you are responsible for making sure that any references and dependencies connected to the deleted node are removed or corrected in the config.xml file.

The script treats all nodes the same. The syntax to run the script is

```
# ./delete-node {node name}
```

so to remove, for example, user 3:

```
# ./delete-node config.users.user3
```

The delete-note script

```
# !/bin/bash
# User must provide the node to be removed. eg "config.users.user1"
# Usage: delete-node {full node path}
if [ $# != 1 ]
then
        echo "Wrong number of arguments"
        echo "Usage: delnode {full '.' delimited node path}"
        exit 2
fi
# test for spaces
TEMP=`echo "$1" | sed 's/.* .*/N/'`
if [ "$TEMP" = "N" ]
then
        echo "Wrong input format"
        echo "Usage: delnode {full '.' delimited node path}"
        exit 2
fi
# testing if node exists
TEMP=`config -g config | grep "$1"`
if [ -z "$TEMP" ]
then
```

```
        echo "Node $1 not found"
        exit 0
fi
# LASTFIELD: last field in the node path. eg "user1"
# ROOTNODE: upper level of the node. eg "config.users"
# NUMBER: integer value extracted from LASTFIELD e.g. "1"
# TOTALNODE: node name for the total e.g. "config.users.total"
# TOTAL: value of the total number of items before deleting eg "3"
# NEWTOTAL: modified total i.e. TOTAL-1
# CHECKTOTAL checks if TOTAL is the actual total items in .xml
LASTFIELD=${1##*.}
ROOTNODE=${1%.*}
NUMBER=`echo $LASTFIELD | sed 's/^[a-zA-Z]*//g'`
TOTALNODE=`echo ${1%.*} | sed 's/\(.*\)/\1.total/'`
TOTAL=`config -g $TOTALNODE | sed 's/.* //'`
NEWTOTAL=$[ $TOTAL -1 ]
# Make backup copy of config file
cp /etc/config/config.xml /etc/config/config.bak
echo "backup of /etc/config/config.xml saved in /etc/config/config.bak"
if [ -z $NUMBER ] # test whether a singular node is being \
                  # deleted e.g. config.sdt.hosts
then
        echo "Deleting $1"
        config -d "$1"
        echo Done
        exit 0
elif [ $NUMBER = $TOTAL ] # Test if only one item exists
then
        echo "only one item exists"
        # Deleting node
        echo "Deleting $1"
        config -d "$1"
        # Modifying item total.
        config -s "$TOTALNODE=0"
        echo Done
        exit 0
elif [ $NUMBER -lt $TOTAL ] # more than one item exists
then
        # Modify the users list so user numbers are sequential
```

```
        # by shifting the users into the gap one at a time...
        echo "Deleting $1"
        LASTFIELDTEXT=`echo $LASTFIELD | sed 's/[0-9]//g'`
        CHECKTOTAL=`config -g $ROOTNODE.$LASTFIELDTEXT$TOTAL`
if [ -z "$CHECKTOTAL" ]
then
        echo "WARNING: "$TOTALNODE" greater than number of items"
 fi
COUNTER=1
while [ $COUNTER != $((TOTAL-NUMBER+1)) ]
do
        config -g $ROOTNODE.$LASTFIELDTEXT$((NUMBER+COUNTER)) \
        | while read LINE
         do
                config -s \
                "`echo "$LINE" | sed -e "s/$LASTFIELDTEXT$((NUMBER+ \
                COUNTER))/$LASTFIELDTEXT$((NUMBER+COUNTER-1))/" \
                -e 's / /=/'`"
        done
        let COUNTER++
done
        # deleting last user
        config -d $ROOTNODE.$LASTFIELDTEXT$TOTAL
        # Modifying item total.
        config -s "$TOTALNODE=$NEWTOTAL"
         echo Done
        exit 0
else
        echo "error: item being deleted has an index greater than total items. Increase the total count variable."
        exit 0
fi
```

16.1.6 POWER CYCLE ANY DEVICE UPON A PING REQUEST FAILURE

The ping-detect script is designed to run specified commands when a monitored host stops responding to ping requests.

The first parameter taken by the ping-detect script is the hostname or IP address of the device to ping. Any other parameters are then regarded as a command to run whenever the ping to the host fails. ping-detect can run any number of commands.

Below is an example using ping-detect to power cycle an RPC (PDU) outlet whenever a specific host fails to respond to a ping request. ping-detect is run from /etc/config/rc.local to make sure that the monitoring starts whenever the system boots.

We assume we have a serially controlled RPC connected to port01 on a console server and have a router powered by outlet 3 on the RPC and the router has an internal IP address of 192.168.22.2. The following instructions will show you how to continuously ping the router and, when the router fails, to respond to a series of pings, the console server will send a command to RPC outlet 3 to power cycle the router, and write the current date/time to a file.

- Copy the ping-detect script to /etc/config/scripts/ on the console server.
- Open /etc/config/rc.local using vi (or another text editor).
- Add the following line to rc.local:

```
/etc/config/scripts/ping-detect 192.168.22.2 /bin/bash -c \
"pmpower -l port01 -o 3 cycle && date" > /tmp/output.log &
```

The above command will cause the ping-detect script to continuously ping the host at 192.168.22.2, which is the router. If the router crashes it will no longer respond to ping requests. If this happens, the two commands pmpower and date will run. The output from these commands is sent to the file /tmp/output.log so that we have some kind of record. The ping-detect is also run in the background using the &.

The rc.local script is only run by default when the system boots. You can manually run the rc.local script or the ping-detect script if desired.

The above is just one example of using the ping-detect script. The idea of the script is to run any number of commands when a specific host stops responding to ping requests. Here are details of the ping-detect script itself.

The ping-detect script

```
# !/bin/sh
# Usage: ping-detect HOST [COMMANDS...]
# This script takes 2 types of arguments: hostname/IPaddress
# to ping, and the commands to run if the ping fails 5 times
# in a row. This script can only take one host/IPaddress per
# instance. Multiple independent commands can be sent to the
# script. The commands will be run one after the other.
#
# PINGREP is the entire reply from the ping command
# LOSS is the percentage loss from the ping command
# $1 must be the hostname/IPaddress of device to ping
# $2... must be the commands to run when the pings fail.
COUNTER=0
TARGET="$1"
shift
```

```
# loop indefinitely:
while true
do
        # ping the device 10 times
        PINGREP=`ping -c 10 -i 1 "$TARGET" `
        # get the packet loss percentage
        LOSS=`echo "$PINGREP" | grep "%" | \
        sed -e 's/.* \([0-9]*\)% .*/\1/'`
if [ "$LOSS" -eq "100" ]
then
        COUNTER=`expr $COUNTER + 1`
else
        COUNTER=0
        sleep 30s
fi
if [ "$COUNTER" -eq 5 ]
then
        COUNTER=0
        "$@"
        sleep 2s
fi
done
```

16.1.7 RUNNING CUSTOM SCRIPTS WHEN A CONFIGURATOR IS INVOKED

A configurator is responsible for reading the values in /etc/config/config.xml and making the appropriate changes live. Some changes made by the configurators are part of the Linux configuration itself, such as user passwords or ipconfig.

Currently there are nineteen configurators, each one responsible for a specific group of config. For example, the users configurator makes the user configurations in the config.xml file live. To see all the available configurators, type the following from a command line prompt:

```
# config
```

When a change is made using the Management Console web GUI, the appropriate configurator is automatically run. This can be problematic. If another user or administrator makes a change using the Management Console, the configurator could possibly overwrite any custom CLI/linux configurations you may have set.

The solution is to create a custom script that runs after each configurator has run. So after each configurator runs, it will check whether that appropriate custom script exists. You can then add any commands to the custom script and they will be invoked after the configurator runs.

The custom scripts must be in the correct location:

```
/etc/config/scripts/config-post-
```

To create an alerts custom script:

```
# cd /etc/config/scripts
# touch config-post-alerts
# vi config-post-alerts
```

This script could be used to recover a specific backup config or overwrite a config or make copies of config files etc.

16.1.8 BACKING-UP THE CONFIGURATION AND RESTORING USING A LOCAL USB STICK

The /etc/scripts/backup-usb script has been written to save and load custom configuration using a USB flash disk. Before saving a configuration locally, you must prepare the USB storage device for use. To do this, disconnect all USB storage devices except for the storage device you wish to use.

Usage:

```
/etc/scripts/backup-usb COMMAND [FILE]
```

TABLE 16-1. USB BACKUP COMMANDS

OPTION	DESCRIPTION
check-magic	Check volume label
set-magic	Set volume label
save file	Save configuration to USB
delete file	Delete a configuration tarball from USB
list	List available config backups on USB
load file	Load a specific config from USB
load-default	Load the default configuration
set-default file	Set which file becomes the default

The first thing to do is to check if the USB disk has a label:

```
# /etc/scripts/backup-usb check-magic
```

If this command returns Magic volume not found, then run the following command:

```
# /etc/scripts/backup-usb set-magic
```

To save the configuration:

```
# /etc/scripts/backup-usb save config-20May
```

To check if the backup was saved correctly:

```
# /etc/scripts/backup-usb list
```

If this command does not display * config-20May then there was an error saving the configuration.

The set-default command takes an input file as an argument and renames it to default.opg. This default configuration remains stored on the USB disk. The next time you want to load the default config, it will be sourced from the new default.opg file. To set a config file as the default:

```
# /etc/scripts/backup-usb set-default config-20May
```

To load this default:

```
# /etc/scripts/backup-usb load-default
```

To load any other config file:

```
# /etc/scripts/backup-usb load {filename}
```

The /etc/scripts/backup-usb script can be executed directly with various commands or called from other custom scripts you may create. However it is recommended that you do not customize the /etc/scripts/backup-usb script itself.

16.1.9 BACKING-UP THE CONFIGURATION OFF-BOX

If you do not have a USB port on your console server, you can back up the configuration to an off-box file. Before backing up, you need to arrange a way to transfer the backup off-box. This could be via an NFS share, a Samba (Windows) share to USB storage or copied off-box via the network. If backing up directly to off-box storage, make sure it is mounted.

/tmp is not a good location for the backup except as a temporary location before transferring it off-box. The /tmp directory will not survive a reboot. The /etc/config directory is not a good place either: it will not survive a restore.

Backup and restore should be done by the root user to ensure correct file permissions are set. The config command is used to create a backup tarball:

```
# config -e <Output File>
```

The tarball will be saved to the indicated location. It will contain the contents of the /etc/config/ directory in an uncompressed and unencrypted form.

Example nfs storage:

```
# mount -t nfs 192.168.0.2:/backups /mnt # config -e /mnt/ \
les1716a-r2.config
# umount/mnt/
```

Example transfer off-box via scp:

```
# config -e /tmp/cm7116.config
# scp /tmp/cm7116.config username@192.168.0.2:/backups
```

The config command is also used to restore a backup:

```
# config -i <Input File>
```

This will extract the contents of the previously created backup to /tmp, and then synchronize the /etc/config directory with the copy in /tmp.

One problem that can crop up here is that there is not enough room in /tmp to extract files to. The following command will temporarily increase the size of /tmp:

```
# mount -t tmpfs -o remount,size=2048k tmpfs /var
```

If restoring to either a new unit or one that has been factory defaulted, it is important to make sure that the process generating SSH keys is either stopped or completed before restoring configuration. If this is not done, then a mix of old and new keys may be put in place.

SSH uses these keys to avoid man-in-the-middle attacks, logging in may be disrupted.

16.2 ADVANCED PORTMANAGER

Black Box's portmanager manages console server serial ports. It routes network connections to serial ports, checks permissions, and monitors and logs all data flowing to and from ports.

16.2.1 PORTMANAGER COMMANDS

pmshell

The pmshell command behaves similarly to standard tip or cu commands, but all serial port access is directed via the portmanager.

For example, to connect to port 8 via the portmanager:

pmshell -l port08

pmshell commands

Once connected, the pmshell command supports a subset of the ~ escape commands that tip and cu support. For SSH you must prefix the escape with an additional ~ character. That is, over SSH use the ~~ escape sequence.

Firmware v3.5.2 and later includes the pmshell chooser escape command. You can now use ~m (or ~~m over SSH) from connected serial port to drop back to pmshell.

For console servers running firmware v3.11.0 and later, pmshell has a set of key sequences built in to access things like the power menu, return to the serial port selection menu and so on.

TABLE 16-2. PMSHELL KEY SEQUENCES

COMMAND	SSH	NOTES
~b	~~b	Generates a BREAK on the connected-to serial port.
~h	~~h	Generates a history on the connected-to serial port. Depends on port logging being enabled.
~p	~~p	Opens the power menu for the connected-to serial port. Said port must be configured for an RPC.
~m	~~m	Connect to the port menu. Goes back to the serial port selection menu.
~?	~~?	Shows the pmshell help message.
~	~~	Quits pmshell.
~?	~~?	Sets RTS to [x].
~signals	~~signals	Shows all signals: 3DSR=1, DTR=1, CTS=1, RTS=1, DCD=0.
~getline	~~getline	Reads a line of text from the serial port.

Extra controls (key sequences) can be added to the built in set of key sequences and can be configured per serial port. You can have all ports behave the same or selectively add control sequences to ports. The controls can be different from port to port for the same function.

For example, you could configure pmshell such that when you are using serial port 2, pressing Ctrl+p would take you straight to the power menu for that port.

The pmshell control commands are configurable only via the command line.

NOTE: The pmshell help message is not updated with information about any custom control command keys that are configured.

There is a helper script which will configure a control command on a range of serial ports to eliminate the cumbersome task of entering the configuration command for every port. You will still need to use this script once per control function (see below), but there are only six of these.

TABLE 16-3. HELPER SCRIPT

PER PORT CONTROL COMMAND CONFIG PARAMETERS	NOTES
config.ports.portX.ctrlcode.break	Generates a BREAK.
config.ports.portX.ctrlcode.portlog	View history
config.ports.portX.ctrlcode.power	open power menu
config.ports.portX.ctrlcode.chooser	connect to port menu
config.ports.portX.ctrlcode.quit	exit pmshell
config.ports.portX.ctrlcode.help	show help message

As an example, to configure Ctrl+p to open the power menu when using serial port 3, enter the following in the console server's command shell:

config -s config.ports.port3.ctrlcode.power=16

killall -HUP portmanager

The first command sets the power menu command to listen for Ctrl+p. Decimal 16 is the character code sent when you press Ctrl+p in the serial port session (see the control codes table immediately below).

The second command — killall -HUP portmanager — tells portmanager to reload the configuration so that the new control code will take effect. Rebooting the device would also work.

There is a script to set serial control codes on a range of ports so that bulk port configuration can be performed more easily. For example to set the power menu control code to CTRL-P (keycode 16) on ports 4 to 10 inclusive, enter the following at the command line:

/etc/scripts/set-serial-control-codes 4 10 power 16

This sets the power menu control key to Ctrl+p.

NOTE: If you've not configured anything on a particular serial port in the included range, configuration for that port will be skipped.

TABLE 16-4. POWER MENU CONTROL KEYS

CONTROL CODE	DECIMAL	CONTROL CODE	DECIMAL	CONTROL CODE	DECIMAL
ctrl+a	1	ctrl+j	10	ctrl+s	19
ctrl+b	2	ctrl+k	11	ctrl+t	20
ctrl+c	3	ctrl+l	12	ctrl+u	21
ctrl+d	4	ctrl+m	13	ctrl+v	22
ctrl+e	5	ctrl+n	14	ctrl+w	23
ctrl+f	6	ctrl+o	15	ctrl+x	24
ctrl+g	7	ctrl+p	16	ctrl+y	25
ctrl+h	8	ctrl+q	17	ctrl+z	26
ctrl+i	9	ctrl+r	18	—	—

pmchat

The pmchat command is similar to the standard chat command, but all serial port access is directed via the portmanager.

For example, to run a chat script via the portmanager:

```
# pmchat -v -f /etc/config/scripts/port08.chat < /dev/port08
```

For more information on using chat and pmchat, you should consult the utility's manual page, via the man chat command on any Linux or UNIX (including macOS) system, or via the web, including at https://linux.die.net/man/8/chat.

pmusers

The pmusers command is used to query the portmanager for active user sessions.

For example, to detect which users are currently active on which serial ports:

```
# pmusers
```

This will output nothing if there are no active users currently connected to any ports. If users are connected, it will respond with a sorted list of usernames per active port. For example:

```
Port 1:
        user1
        user2
Port 2:
        user1
Port 8:
        user2
```

The above output indicates that a user named user1 is actively connected to ports 1 and 2, while user2 is connected to both ports 1 and 8.

With firmware v3.11 and later the pmusers command is extended with the --disconnect option. This allows an administrator or root to disconnect console server sessions from the command line. The following connection types can be disconnected:

telnet

SSH

Raw TCP

Unauthorized Telnet

The --disconnect option cannot disconnect an RFC2217 session.

If the --disconnect option is specified, the pmusers command goes into disconnect mode where you can specify users with -u and ports with -l (by label) or -n (by name).

By default, the command will prompt the user before actually disconnecting the matching sessions. This can be overriden with the --no-prompt argument.

Some example pmuser sessions:

```
# pmusers --disconnect
Disconnect all users from all ports? (y/n)
y
5 sessions were disconnected
```

```
# pmusers --disconnect -u robertw
Disconnect user robertw from all ports? (y/n)
y
1 session was disconnected

# pmusers --disconnect -u robertw -n 5
Disconnect user robertw from port 5 (BranchRouter01)? (y/n)
y
No sessions were disconnected

# pmusers --disconnect -n 5
Disconnect all users from port 5 (BranchRouter01)? (y/n)
y
2 sessions were disconnected

# pmusers --disconnect -u robertw -u pchunt -n 4 -n 6
Disconnect users robertw, pchunt from ports 4, 6? (y/n)
y
10 sessions were disconnected

# pmusers --disconnect -u tester --no-prompt
No sessions were disconnected
```

portmanager dæmon

There is normally no need to stop and restart the dæmon. To restart the dæmon normally, just run the command:

```
# portmanager
```

Supported command line options are:

TABLE 16-5. SUPPORTED COMMAND OPTIONS

OPTION	PURPOSE
-nodaemon	Force portmanager to run in the foreground
--loglevel={debug | info |warn | error | alert}	Set the level of debug logging
-c /etc/config/portmanager.conf	Change which configuration file it uses

signals

Sending a SIGHUP signal to the portmanager will cause it to re-read its configuration file.

16.2.2 EXTERNAL SCRIPTS AND ALERTS

The portmanager has the ability to execute external scripts on certain events.

When a port is opened by the portmanager

When portmanager opens a port, it attempts to execute /etc/config/scripts/portXX.init (where XX is the number of the port, for example 08). The script is run with STDIN and STDOUT both connected to the serial port.

If the script cannot be executed, portmanager executes /etc/config/scripts/portXX.chat via the chat command on the serial port.

When an alert occurs on a port

When an alert occurs on a port, portmanager attempts to execute /etc/config/scripts/portXX.alert (where XX is the port number, for example 08).

The script is run with STDIN containing the data which triggered the alert, and STDOUT redirected to /dev/null, not to the serial port.

If you wish to communicate with the port, use pmshell or pmchat from within the script.

If the script cannot be executed, the alert will be mailed to the address configured in the system administration section.

When a user connects to any port

If a file called /etc/config/pmshell-start.sh exists it is run when a user connects to a port. It is provided 2 arguments, the Port number and the Username. Here is a simple example:

```
</etc/config/pmshell-start.sh>
#!/bin/sh
PORT="$1"
USER="$2"
echo "Welcome to port $PORT $USER"
</etc/config/pmshell-start.sh>
```

The return value from the script controls whether the user is accepted or not, if 0 is returned (or nothing is done on exit as in the above script) the user is permitted, otherwise the user is denied access.

Here is a more complex script which reads from configuration to display the port label, if available, and denies access to the root user:

```
</etc/config/pmshell-start.sh>
#!/bin/sh
PORT="$1"
USER="$2"
LABEL=$(config -g config.ports.port$PORT.label | cut -f2- -d' ')
if [ "$USER" == "root" ]; then
        echo "Permission denied for Super User"
        exit 1
fi
if [ -z "$LABEL" ]; then
        echo "Welcome $USER, you are connected to Port $PORT"
```

```
else
        echo "Welcome $USER, you are connected to Port $PORT ($LABEL)"
fi
</etc/config/pmshell-start.sh>
```

16.3 RAW ACCESS TO SERIAL PORTS

16.3.1 ACCESS TO SERIAL PORTS

You can use tip and stty to completely bypass the portmanager and have raw access to the serial ports.

When you run tip on a portmanager-controlled port, portmanager closes that port, and stops monitoring it until tip releases control of it.

With stty, the changes made to the port only persist until that port is closed and opened again. Using stty for more than initial debugging of a serial connection is not recommended.

If you want to use stty to configure a port, you can put stty commands in /etc/config/scripts/portXX.init which gets run whenever portmanager opens the port Otherwise, any setup you do with stty will be lost when portmanager opens the port.

NOTE: portmanager sets things back to its config rather than using whatever is on the port, so the port is in a known good state, and will work, no matter what things are done to the serial port outside of portmanager.

16.3.2 ACCESSING THE CONSOLE MODEM PORT

Console dial-in is handled by mgetty, with automatic PPP login extensions. mgetty is a smart getty replacement, designed for Hayes-compatible data and data/fax modems.

mgetty knows about modem initialization, manual modem answering (so your modem doesn't answer if the machine isn't ready), and UUCP locking (so you can use the same device for dial-in and dial-out). mgetty provides extensive logging facilities. All standard mgetty options are supported.

Modem initialization strings

To override the standard modem initialization string either use the Management Console (see chapter 4) or the command line config tool (see chapter 14).

Enabling boot messages on the console

If you are not using a modem on the DB9 console port and instead wish to connect to it directly via a Null Modem cable you may want to enable verbose mode allowing you to see the standard linux start-up messages. This can be achieved with the following commands:

```
# /bin/config --set=config.console.debug=on # /bin/config \
--run=console # reboot
```

If at some point in the future you chose to connect a modem for dial-in out-of-band access the procedure can be reversed with the following commands:

```
# /bin/config --del=config.console.debug
# /bin/config --run=console # reboot
```

16.4 IP FILTERING

The console server uses the iptables utility to provide a stateful firewall of LAN traffic.

By default, rules are automatically inserted to allow access to enabled services, and serial port access via enabled protocols. The commands which add these rules are in configuration files:

/etc/config/fw.rules

This is an executable shell script which is run whenever the LAN interface is brought up and whenever modifications are made to the iptables configuration as a result of CGI actions or the config command line tool.

The basic steps performed are as follows:

- the running iptables configuration is erased, per-interface.
- other standard system chains are installed.
- fall-through Block rules (default deny) are installed.
- Serial & Network > Services policies are installed in per-interface chains.
- Custom Serial & Network > Firewall rules are inserted at the top of the rule sets, taking priority over any other configuration

For further firewall customization, extra rules can be persisted by creating a file at /etc/config/scripts/firewall-post containing iptables commands to amend the firewall policy.

Thorough documentation regarding iptables is available at the Linux netfilter website, at https://netfilter.org/documentation/.

16.5 SNMP STATUS REPORTING

Console servers contain an SNMP Service — snmpd — which can provide status information on demand. snmpd is an SNMP agent which binds to a port and awaits requests from SNMP management software. Upon receiving a request, it processes the request(s), collects the requested information and/or performs the requested operation(s) and returns the information to the sender.

NOTE: Initially, only advanced console server models were equipped with an SNMP Service. With firmware v3.0 and later this support was extended to all console servers. Also the MIBS were extended (and renamed for compliance) with this firmware release.

Console servers can also be configured to send SNMP traps or messages to multiple remote SNMP Network Managers on defined trigger events. See chapter 8 for configuration details.

16.5.1 RETRIEVING STATUS INFORMATION USING SNMP

Console servers can provide serial and device status information through SNMP. This includes

- Serial port status
- Active users
- Remote Power Control (RPC) and Power Distribution Unit (PDU) status
- Environmental Monitoring Device (EMD) status
- Signal alert status
- Environmental alert status and
- UPS alert status

The MIBs in your console server are located in /etc/snmp/mibs. They include:

TABLE 16-6. MIBS

MIB	NOTES
OG-STATUS-MIB	Contains serial and connected device status information for snmpstatusd and snmpalertd.
OG-STATUSv2-MIB	This MIB contains extended status and alerts.
OG-SMI-MIB	Enterprise structure of management information.
OGTRAP-MIB	SMIv1 traps from old MIBS as smilint will not let SMIv1 structures coexist with SMIv2.
OGTRAPv2-MIB	Updated traps.

16.5.2 CHECK FIREWALL RULES

- Navigate to System > Services.
- Check the SNMP dæmon checkbox for the required interface.

  This allows SNMP requests through the specified interface's firewall.

16.5.3 CHECK FIREWALL RULES

Console servers support different SNMP versions including SNMPv1, SNMPv2c and SNMPv3.

Although an industry standard, SNMP brings with it a variety of security concerns. For example, SNMPv1 and SNMPv2c offer no inherent privacy, while SNMPv3 is susceptible to man-in-the-middle attacks. Recent IETF developments suggest tunnelling SNMP over widely accepted technologies such as SSH (Secure Shell) or TLS (Transport Layer Security) rather than relying on a less mature security systems such as SNMPv3's USM (User-based Security Model).

Additional information regarding SNMP security issues and SNMPv3 can be found at http://net-snmp.sourceforge.net/wiki/index.php/TUT:Security.

- Navigate to Alerts & Logging > SNMP.

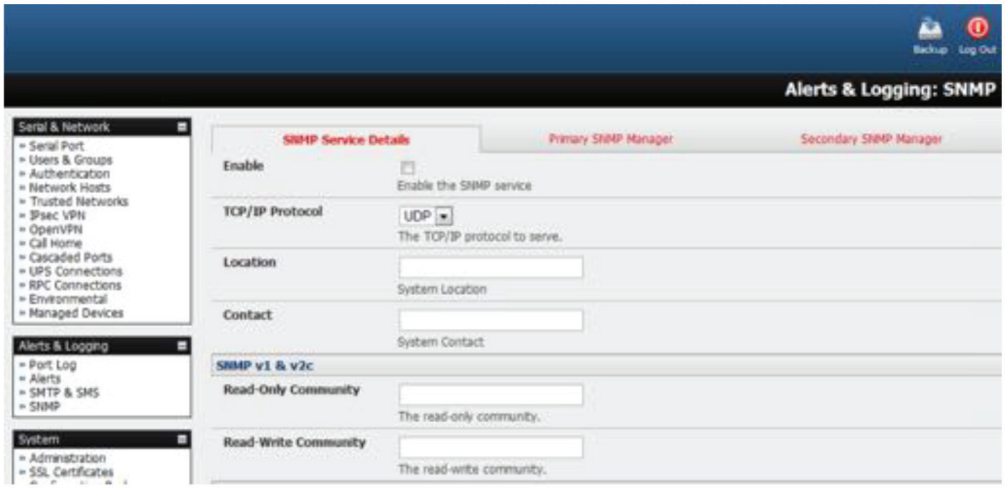


FIGURE 16-1. ALERTS & LOGGING: SNMP SCREEN

The SNMP Service Details tab shows by default. This tab controls aspects of the SNMP service including security level. It also manages requests from external agents for Black Box status information.

- Check the Enable the SNMP Service to start the SNMP service.

  SNMP is disabled by default.

- Select either UDP or TCP for the TCP/IP Protocol.

  UDP is the recommended protocol and is selected by default. TCP should only be used in special cases, such as when Port Forwarding SNMP requests/responses to or from the Black Box device is required.

- Complete the Location and Contact fields.

  The Location field should describe the physical location of the Black Box and will be used in response to requests for the SNMPv2-MIB::sysLocation.0 of the device.

  The Contact field refers to the person responsible for the Black Box such as the System Administrator and will be used in response to requests as follows: SNMPv2-MIB::sysContact.0.

- Enter the Read-Only Community and Read-Write Community.

  This is required for SNMP v1 & v2c only. The Read-Only Community field is used to specify the SNMPv1 or SNMPv2c community that will be allowed read-only (GET and GETNEXT) access. This must be specified in order for both versions to become enabled. The Read-Write Community field is used to specify the SNMPv1 or SNMPv2c community that will be allowed read-write (GET, GETNEXT and SET) access.

- Configure SNMPv3, if required.

  SNMPv3 provides secure SNMP operations through the use of USM (User-based Security Model). It offers various levels of security including user-based authentication and basic encryption.

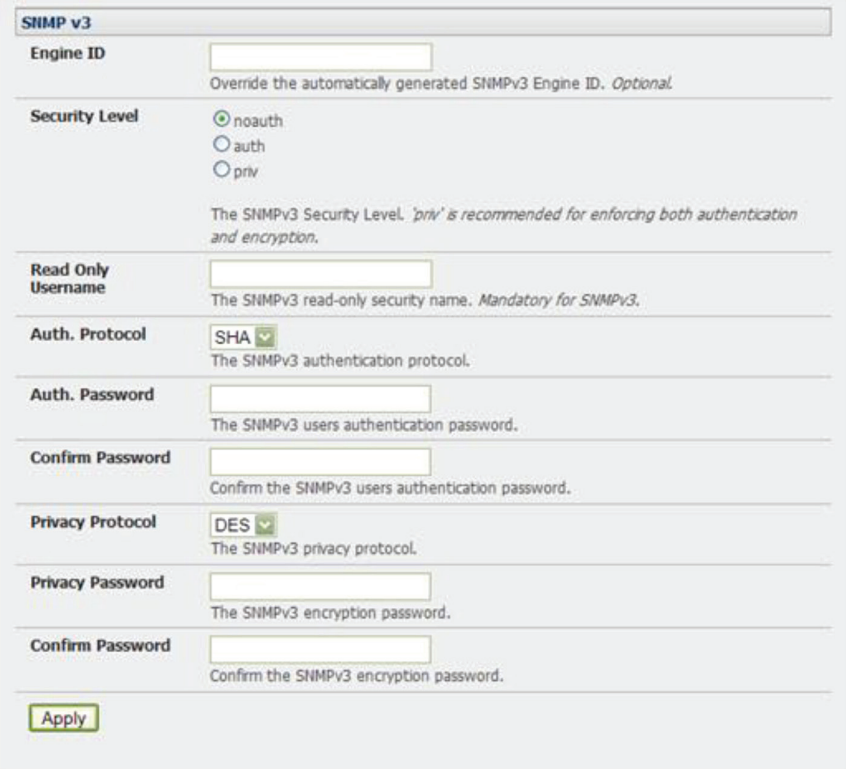


FIGURE 16-2. SNMPV3 SCREEN

- Enter an Engine ID if required.

  Engine ID is used to localize the SNMPv3 user. It will be automatically generated from a Network Interface (eth0) hardware address, if left blank, or must be entered as a hex value (for example, 0x01020304).
- Specify the Security Level.

TABLE 16-7. SECURITY LEVELS

SECURITY LEVEL	NOTES
noauth	No authentication or encryption required. This is the minimum security level.
auth	Authentication will be required but encryption is not enforced. An authentication protocol (SHA or MD5) and password will be required.
priv	Enforces encryption use. This is the highest level of security and requires an encryption protocol (DES or AES) and password in addition to the authentication protocol and password.

- Enter the Read Only Username.

  This field is mandatory when configuring the console server for SNMPv3.
- For a Security Level of auth, set the Auth Protocol (SHA or MD5) and the Auth Password.

  A password of at least 8 characters is required.
- For a Security Level of priv, set the Privacy Protocol (DES or AES) and the Privacy Password.

  AES is recommended. A password of at least 8 characters is required.
- Click Apply.
- Setup serial ports and devices as per requirements such as UPS, RPC/PDU and EMD.
- Copy the mibs from /etc/snmp/mibs on the console server to a local directory using scp or Winscp. For example:

  scp root@im4004:/etc/snmp/mibs/*
- Using the snmpwalk and snmpget commands, status information can be retrieved from any console server. For example:

  snmpwalk -Oa -v1 -M .:/usr/share/snmp/mibs -c public im4004 OG-STATUS-MIB::ogStatus

  snmpget -Oa -v1 -M .:/usr/share/snmp/mibs -c public im4004 OG-STATUSMIB::ogSerialPortStatusSpeed.2

noauth

snmpwalk -Oa –v3 –l noAuthNoPriv –u readonlyusername -M .:/usr/share/snmp/mibs im4004 OG-STATUS-MIB::ogStatus

auth

snmpwalk -Oa –v3 –l authNoPriv –u readonlyusername –a SHA –A “authpassword” -M .:/usr/share/snmp/mibs im4004 OG-STATUS-MIB::ogStatus

priv

snmpwalk -Oa –v3 –l authNoPriv –u readonlyusername –a SHA –A “authpassword” –x DES –X “privpassword” -M .:/usr/share/snmp/mibs im4004 OG-STATUS-MIB::ogStatus

TABLE 16-8. SNMP ARGUMENTS

SNMP ARGUMENT	PURPOSE
-l	security level
-u	security name or read-only username
-a	authentication protocol: SHA or MD5
-A	authentication password
-x	privacy protocol: DES or AES
-X	privacy password

A mib browser can explore the Black Box enterprise MIB structure.

16.5.4 ADDING MULTIPLE REMOTE SNMP MANAGERS

You can add multiple SNMP servers for alert traps. Add the first and second SNMP servers using the Management Console (see Chapter 8) or the command line config tool. Further SNMP servers must be added manually using config.

Log in to the console server's command line shell as root or an admin user.

- Set the SNMP Manager Address field:
  config –set="config.system.snmp.address3=w.x.y.z"
  replacing w.x.y.z with the IP address or hostname.
- Set the Manager Trap Port field:
  config --set=”config.system.snmp.trapport3=162”
  replacing 162 with the TCP/UDP port number
- Set the SNMP Manager Protocol field:
  config --set="config.system.snmp.protocol3=UDP"
  or
  config --set="config.system.snmp.protocol3=TCP"
- Set the SNMP Manager Version field:
  config --set="config.system.snmp.version3=3"
- Set the SNMP Manager v1 & v2c community field:
  config --set="config.system.snmp.community3=public"
- Set the SNMP Manager v3 Engine ID field:
  config –set="config.system.snmp.engineid3=0x8000000001020304"
  replacing 0x8000000001020304 with the hex Engine-ID.
- Set the SNMP Manager v3 Security Level field:
  config --set="config.system.snmp.seclevel3=noAuthNoPriv"
  or
  config --set="config.system.snmp.seclevel3=authNoPriv"
  or
  config --set="config.system.snmp.seclevel3=authPriv"

- Set the SNMP Manager v3 Username field:

  config --set="config.system.snmp.username3=username"
- Set the SNMP Manager v3 Auth. Protocol and password fields:

  config –set="config.system.snmp.authprotocol3=SHA"

  or

  config --set="config.system.snmp.authprotocol3=MD5"

  config --set="config.system.snmp.authpassword3=password 1"
- To set the SNMP Manager v3 Privacy Protocol and password fields:

  config –set="config.system.snmp.privprotocol3=AES"

  or

  config –set="config.system.snmp.privprotocol3=DES"

  config --set="config.system.snmp.privpassword3=password 2"
- Once the fields are set, apply the configuration with the following command:

  config --run snmp

You can add a third or more SNMP servers by incrementing the 2 in the above commands. For example, config.system.snmp.protocol3, config.system.snmp.address3, etc.

16.6 SECURE SHELL (SSH) PUBLIC KEY AUTHENTICATION

This section covers the generation of public and private keys in a Linux and Windows environment and configuring SSH for public key authentication. The steps to use in a Clustering environment are:

- generate a new public and private key pair.
- upload the keys to the master and to each slave console server.
- fingerprint each connection to validate.

16.6.1 SSH OVERVIEW

Popular TCP/IP applications such as telnet, rlogin, ftp, and others transmit their passwords unencrypted. Doing this across pubic networks like the Internet can have catastrophic consequences. It leaves the door open for eavesdropping, connection hijacking, and other network-level attacks.

Secure Shell (SSH) is a program to log into another computer over a network, to execute commands in a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over insecure channels.

OpenSSH, the de facto open source SSH application, encrypts all traffic (including passwords) to effectively eliminate these risks. Additionally, OpenSSH provides a myriad of secure tunneling capabilities, as well as a variety of authentication methods.

OpenSSH is the port of OpenBSD's excellent OpenSSH[0] to Linux and other versions of Unix. OpenSSH is based on the last free version of Tatu Ylonen’s sample implementation with all patent-encumbered algorithms removed (to external libraries), all known security bugs fixed, new features reintroduced and many other clean-ups.

The only changes in the Black Box SSH implementation are:

- PAM support.
- EGD[1]/PRNGD[2] support and replacements for OpenBSD library functions that are absent from other versions of UNIX.

- The config files are now in /etc/config/. For example:

  /etc/config/sshd_config not /etc/sshd_config

  /etc/config/ssh_config not /etc/ssh_config

  /etc/config/users/<username>/.ssh / not /home/<username>/.ssh/

16.6.2 GENERATING PUBLIC KEYS (LINUX)

To generate new SSH key pairs use the Linux ssh-keygen command.

This produces an RSA or DSA public/private key pair. You will be prompted for a path to store the two key files: id_dsa.pub (the public key) and id_dsa (the private key). For example:

```
$ ssh-keygen -t [rsa|dsa]
Generating public/private [rsa|dsa] key pair.
Enter file in which to save the key (/home/user/.ssh/id_[r|dsa]):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_[r|dsa].
Your public key has been saved in /home/user/.ssh/id_[r|dsa].pub.
The key fingerprint is:
28:aa:29:38:ba:40:f4:11:5e:3f:d4:fa:e5:36:14:d6 user@server
$
```

It is advisable to create a new directory to store your generated keys. It is also possible to name the files after the device they will be used for. For example:

```
$ mkdir keys
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key: ~/keys/control_room
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ~/keys/control_room
Your public key has been saved in ~/keys/control_room.pub.
The key fingerprint is:
28:aa:29:38:ba:40:f4:11:5e:3f:d4:fa:e5:36:14:d6 user@server
$
```

There must be no password associated with the keys. If there is a password, Black Box devices will have no way to supply it at runtime.

Full documentation for the ssh-keygen command can be found at http://man.openbsd.org/OpenBSD-current/man1/ssh-keygen.1.

16.6.3 INSTALLING THE SSH PUBLIC & PRIVATE KEYS (CLUSTERING)

For console servers the keys can be uploaded through the web interface, on the System > Administration page.

FIGURE 16-3. SYSTEM > ADMINISTRATION SCREEN

This enables you to upload stored RSA or DSA Public Key pairs to the master and apply the authorized key to the slave as documented in chapter 5. Once complete you then proceed to Fingerprinting as documented below.

16.6.4 INSTALLING THE SSH PUBLIC & PRIVATE KEYS (CLUSTERING)

Alternately, the public key can be installed on the unit remotely from the linux host with the scp utility.

Assumptions:

- the Management Console username is fred.
- the console server IP address is 192.168.0.1 (a console server's default private IP address).
- the public key is stored on the Linux- or UNIX-based system in ~/.ssh/id_dsa.pub.

Given this, run the following command from the Linux- or UNIX-based system:

```
scp ~/.ssh/id_dsa.pub \
root@192.168.0.1:/etc/config/users/fred/.ssh/authorized_keys
```

This copies the file to the console server but doesn't set ownership as required. The authorized_keys file on the console server needs to be owned by fred. To affect this, login to the Management Console as root and run the following command:

```
chown fred /etc/config/users/fred/.ssh/authorized_keys
```

If the console server selected to be the server has only one client device, the authorized_keys file is simply a copy of the public key for that device.

If one or more devices will be clients of the console server, the authorized_keys file will contain copies of all of the public keys.

RSA and DSA keys may be freely mixed in the authorized_keys file. For example, assume we already have one server, called bridge_server, and two sets of keys, for the control_room and the plant_entrance. The following commands 1) show the stored keys and 2) combine two of them into a single file, authorized_keys_bridge_server.

```
$ ls /home/user/keys
```

control_room

control_room.pub

plant_entrance

plant_entrance.pub

$ cat ~/keys/control_room.pub ~/keys/plant_entrance.pub > ~/keys/authorized_keys_bridge_server

More OpenSSH documentation can be found at https://openssh.com/manual.html and http://man.openbsd.org/OpenBSD-current/man1/ssh.1.

16.6.5 GENERATING PUBLIC AND PRIVATE KEYS FOR SSH (WINDOWS)

This section describes how to generate and configure SSH keys using Windows.

The OpenSSH project does not produce a Windows binary. The OpenSSH project's development is entirely focussed on producing 'a very small, secure, and easy to maintain version for the OpenBSD project'.

The versions of OpenSSH that ship on other Unix- and Unix-like operating systems are managed and produced by the OpenSSH Portability Team.

As of 2016-10, and despite Microsoft announcing 'the PowerShell team will support and contribute to the OpenSSH community… to deliver the PowerShell and Windows SSH solution', there is no Windows version of the current OpenSSH release.

Consequently, Simon Tatham's long-standing SSH client for Windows, PuTTY, which includes the key generator, PuTTYgen.exe, is used in the following procedure.

Before beginning, make sure you have the most recent PuTTYgen release installed. PuTTYgen is available for download from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html.

This procedure also requires the current version of WinSCP — a Windows-equivalent to the scp utliltiy — be installed. WinSCP is available for download from https://winscp.net/.

- Create a new user from the Black Box Management Console.

  The following example uses a user called testuser. This user must be a member of the users group.
- If you do not already have a public/private key pair generate them now using PuTTYgen.
- Launch PuTTYgen.exe.
- Select the desired key type — SSH2 DSA — in the Parameters section.

  You may use RSA or DSA.
- Leave the passphrase field blank.
- Click Generate.
- As instructed, move the mouse pointer over the blank area of the program in order to create random data used by PUTTYGEN to generate secure keys.

  Key generation occurs once PUTTYGEN has collected sufficient random data.
- Copy the public key data from the Public key for pasting into OpenSSH authorized_keys file section of the PuTTY Key Generator window.
- Launch Notepad (not Microsoft Word or any other word processor).
- Paste the key data into the Notepad window.

  Make sure there is only one line of text in this file.
- Save the Notepad file as authorized_keys.
- Launch WinSCP.

- Copy authorized_keys to the user's home directory on the console server which will be the SSH server.

  For example, if the user's username is testuser, copy the file to

  /etc/config/users/testuser/.ssh/authorized_keys
- From the console server's command line run the following commands to give the file the correct text-format and the correct permissions:

  # dos2unix /etc/config/users/testuser/.ssh/authorized_keys

  # chown testuser /etc/config/users/testuser/.ssh/authorized_keys
- Using WinSCP, copy the local sshd_config file over /etc/config/sshd_config on the console server.

  This ensures public key authentication is enabled.
- Test the public key by logging in to the console server as testuser.
- At the console server's command line type the following:

  # ssh -o StrictHostKeyChecking=no <server-ip>

To automate connection of the SSH tunnel from the client on every power-up you need to make the client's /etc/config/rc.local look like the following:

```
#!/bin/sh
ssh -L9001:127.0.0.1:4001 -N -o \
StrictHostKeyChecking=no testuser@<server-ip> &
```

This will run the tunnel redirecting local port 9001 to the server port 4001.

16.6.6 FINGERPRINTING

Fingerprints are used to ensure you are establishing an SSH session to who you think you are. On the first connection to a remote server you will receive a fingerprint which you can use on future connections.

This fingerprint is related to the host key of the remote server. Fingerprints are stored in ~/.ssh/known_hosts.

- To receive the fingerprint from the remote server, log in to the client as the required user (usually root) and establish a connection to the remote host:

  # ssh rh

  The authenticity of host 'rh (192.168.0.1)' can't be established.

  RSA key fingerprint is 8d:11:e0:7e:8a:6f:ad:f1:94:0f:93:fc:7c:e6:ef:56.

  Are you sure you want to continue connecting (yes/no)?
- Answer yes to accept the key.
- The following message will be returned:

  Warning: Permanently added 'rh,192.168.0.1' (RSA) to the list of known hosts.
- You may be prompted for a password.

  There is no need to log in, however: you have received the fingerprint.
- Press Ctrl-C to cancel the connection.

  If the host key changes you will receive the following warning, and not be allowed to connect to the remote host:

  WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!

  IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!

Someone could be eavesdropping on you right now using a man-in-the-middle attack.

It is also possible that the RSA host key has just been changed:

The fingerprint for the RSA key sent by the remote host is

ab:7e:33:bd:85:50:5a:43:0b:e0:bd:43:3f:1c:a5:f8.

Please contact your system administrator.

Add correct host key in /.ssh/known_hosts to get rid of this message.

Offending key in /.ssh/known_hosts:1

RSA host key for remhost has changed and you have requested strict checking.

Host key verification failed.

If the host key has legitimately changed, it can be removed from the ~/.ssh/known_hosts file and the new fingerprint added. If it has not changed legitimately, this indicates a serious problem that should be investigated immediately.

16.6.7 INSTALLING THE SSH PUBLIC & PRIVATE KEYS (CLUSTERING)

You can apply SSH tunneling when two Black Box console servers are configured for serial bridging.

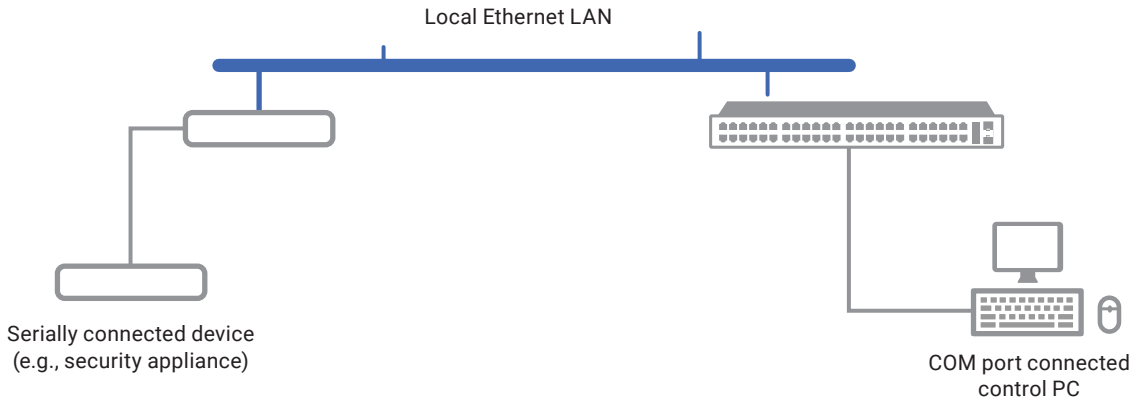


FIGURE 16-4.

As detailed in Chapter 5, the Server console server is setup in Console Server mode with either RAW or RFC2217 enabled and the Client console server is set up in Serial Bridging Mode with the Server Address, and Server TCP Port (4000 + port for RAW or 5000 + port # for RFC2217) specified:

- Select SSH Tunnel when configuring the Serial Bridging Setting.

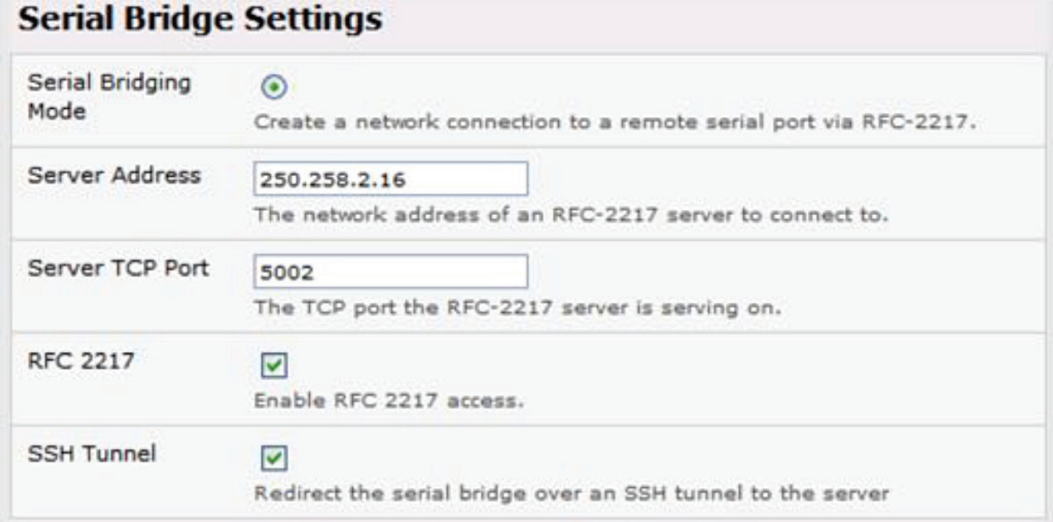


FIGURE 16-5.

- Set up SSH keys for each end of the tunnel and upload these keys to the Server and Client console servers.

Client keys

The first step in setting up ssh tunnels is to generate keys. Ideally, you will use a separate, secure, machine to generate and store all keys to be used on the console servers. However, if this is not ideal to your situation, keys may be generated on the console servers themselves.

It is possible to generate only one set of keys, and reuse them for every SSH session. While this is not recommended, each organization will need to balance the security of separate keys against the additional administration they bring.

Generated keys may be one of two types — RSA or DSA — and it is beyond the scope of this document to recommend one over the other. RSA keys will go into the files id_rsa and id_rsa.pub. DSA keys will be stored in the files id_dsa and id_dsa.pub.

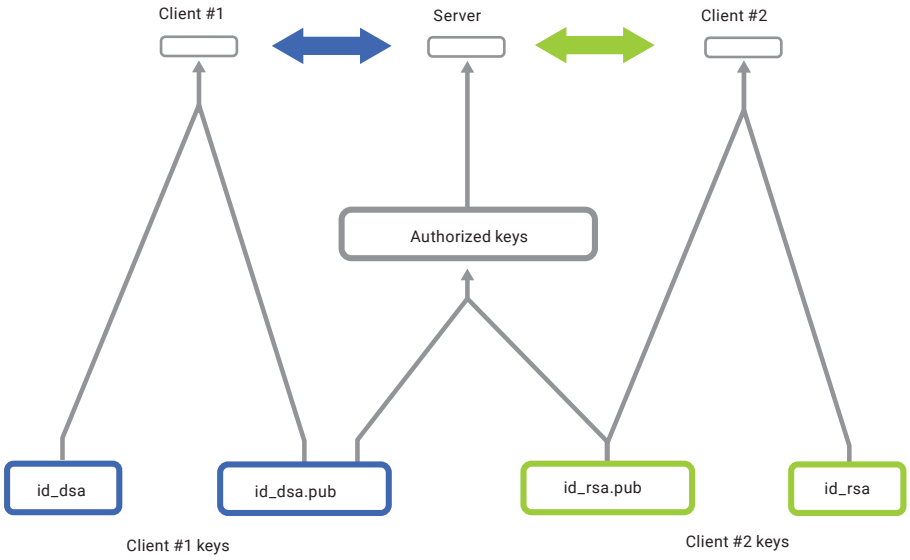


FIGURE 16-6. CLIENT KEYS

For simplicity going forward the term private key will be used to refer to either id_rsa or id_dsa and public key to refer to either id_rsa.pub or id_dsa.pub.

To generate the keys use the ssh-keygen program (part of the OpenSSH suite):

```
$ ssh-keygen -t [rsa|dsa]
Generating public/private [rsa|dsa] key pair.
Enter file in which to save the key (/home/user/.ssh/id_[r|dsa]):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_[r|dsa].
Your public key has been saved in /home/user/.ssh/id_[r|dsa].pub.
The key fingerprint is:
28:aa:29:38:ba:40:f4:11:5e:3f:d4:fa:e5:36:14:d6 user@server
$
```

It is advisable to create a new directory to store your generated keys. It is also possible to name the files after the device they will be used for. For example:

```
$ mkdir keys
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key: ~/keys/control_room
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ~/keys/control_room
Your public key has been saved in ~/keys/control_room.pub.
The key fingerprint is:
28:aa:29:38:ba:40:f4:11:5e:3f:d4:fa:e5:36:14:d6 user@server
$
```

There must be no password associated with the keys. If there is a password, Black Box devices will have no way to supply it at runtime.

Authorized keys

If the console server selected to be the server has only one client device, the authorized_keys file is simply a copy of the public key for that device.

If one or more devices will be clients of the console server, the authorized_keys file will contain copies of all of the public keys.

RSA and DSA keys may be freely mixed in the authorized_keys file. For example, assume we already have one server, called bridge_server, and two sets of keys, for the control_room and the plant_entrance. The following commands 1) show the stored keys and 2) combine two of them into a single file, authorized_keys_bridge_server.

```
$ ls /home/user/keys
control_room
control_room.pub
plant_entrance
```

plant_entrance.pub

$ cat ~/keys/control_room.pub ~/keys/plant_entrance.pub > ~/keys/authorized_keys_bridge_server

Uploading keys

The keys for the server can be uploaded through the web interface, on the System > Administration page as detailed earlier. If only one client will be connecting, then simply upload the appropriate public key as the authorized keys file. Otherwise, upload the authorized keys file constructed in the previous step.

Each client will then need its own set of keys uploaded through the same page. Take care to ensure that the correct type of keys (DSA or RSA) goes in the correct spots, and that the public and private keys are in the correct spot.

16.6.8 SDT CONNECTOR PUBLIC KEY AUTHENTICATION

SDT Connector can authenticate against a console server using your SSH key pair rather than requiring your to enter your password (that is public key authentication).

To use public key authentication with SDT Connector, first create an RSA or DSA key pair (using ssh-keygen, PuTTYgen or a similar tool) and add the public part of your SSH key pair to the console server.

Next, add the private part of your SSH key pair (this file is typically named id_rsa or id_dsa) to the SDT Connector client:

- Navigate to Edit > Preferences > Private Keys > Add.
- Locate the private key file.
- Click OK.

You do not have to add the public part of your SSH key pair, it is calculated using the private key.

SDT Connector will now use public key authentication when SSH-connecting through the console server. You may have to restart SDT Connector to shut down any existing tunnels that were established using password authentication.

If you have a host behind the console server that you connect to by clicking the SSH button in SDT Connector, you can also configure it for public key authentication.

Essentially what you are using is SSH over SSH. The two SSH connections are entirely separate, and the host configuration is entirely independent of SDT Connector and the console server. You must configure the SSH client that SDT Connector launches (for example Putty or OpenSSH) and the host's SSH server for public key authentication.

16.7 SECURE SOCKETS LAYER (SSL) SUPPORT

Secure Sockets Layer (SSL) is a protocol developed by Netscape for transmitting private documents via the Internet. SSL works by using a private key to encrypt data that's transferred over the SSL connection.

The console server includes OpenSSL. The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. The project is managed by a worldwide community of volunteers that use the Internet to communicate, plan, and develop the OpenSSL toolkit and its related documentation.

OpenSSL is based on the Slay library developed by Eric A Young and Tim J Hudson. The OpenSSL toolkit is licensed under an Apache-style license, which basically means that you are free to get and use it for commercial and non-commercial purposes subject to some simple license conditions. In the console server OpenSSL is used primarily in conjunction with https in order to have secure browser access to the GUI management console across insecure networks.

OpenSSL documentation is available at https://openssl.org/docs/manmaster/apps/openssl.html.

The OpenSSL project itself 'highly recommends' Ivan Ristić's OpenSSL Cookbook, available as a free download from https://feistyduck.com/books/openssl-cookbook/.

16.8 HTTPS

The Management Console UI is served using HTTPS by the built in cherokee webserver.

If your default network address is changed or the unit is to be accessed via a known Domain Name you can use the following steps to replace the default SSL Certificate and Private Key with ones tailored for your new address.

16.8.1 GENERATING AN ENCRYPTION KEY

To create a 1024 bit RSA key with a password issue the following command on the command line of a linux host with the openssl utility installed:

```
# openssl genrsa -des3 -out ssl_key.pem 1024
```

16.8.2 GENERATING A SELF-SIGNED CERTIFICATE WITH OPENSSL

This example shows how to use OpenSSL to create a self-signed certificate on a Linux- or Unix-based system. OpenSSL ships as part of macOS and is available for most Linux distributions via the default package management mechanism.

The OpenSSL project 'does not distribute any code in binary form, and does not officially recommend any specific binary distributions.' The project does, however, maintain a page on its community wiki: https://wiki.openssl.org/index.php/Binaries.

This page lists 3rd-party binaries that are 'stable and can provide continued support for OpenSSL'. Windows users should check here for a suitable binary.

To create a 1024-bit RSA key and a self-signed certificate, issue the following command from the host you have openssl installed on:

```
# openssl req -x509 -nodes -days 1000 -newkey rsa:1024 -keyout \
ssl_key.pem -out ssl_cert.pem
```

You will be prompted to enter a lot of information. Most of it doesn't matter, but the Common Name should be the domain name of your computer (for example, test.Black Box.com).

When you have entered everything, the certificate will be created in a file called ssl_cert.pem.

16.8.3 INSTALLING THE KEY AND CERTIFICATE

The recommended method for copying files securely to the console server unit is with a Secure Copying Protocol client (for example, the shell-based tool: scp).

The scp utility ships with macOS and ships with OpenSSH for most Linux distributions. Windows users can use something like the PSCP command line utility available with PuTTY.

The files created in the steps above can be installed remotely with the scp utility as follows:

```
# scp ssl_key.pem root@<address of unit>:/etc/config/
# scp ssl_cert.pem root@<address of unit>:/etc/config/
```

or, using PSCP:

pscp -scp ssl_key.pem root@<address of unit>:/etc/config/

pscp -scp ssl_cert.pem root@<address of unit>:/etc/config/

PuTTY and the PSCP utility can be downloaded from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html.

Detailed documentation on PSCP can be found at https://the.earth.li/~sgtatham/putty/0.67/htmldoc/Chapter5.html.

16.8.4 LAUNCHING THE HTTPS SERVER

The easiest way to enable the HTTPS server is from the web Management Console.

- Click the appropriate checkbox in Network > Services > HTTPS Server.

  The HTTPS server will now be activated (assuming ssl_key.pem and ssl_cert.pem exist in the /etc/config/).

Alternatively, inetd can be configured to launch the secure fnord server from the command line of the unit as follows.

- Edit the inetd configuration file. From the unit command line:

  # vi /etc/config/inetd.conf
- Append a line:

  443 stream tcp nowait root sslwrap -cert /etc/config/ssl_cert.pem -key /etc/config/ssl_key.pem -exec /bin/httpd /home/httpd"
- Save the file.
- Signal inetd of the configuration change:

  # kill -HUP `cat /var/run/inetd.pid`

The HTTPS server should now be accessible from a web client at a URL similar to this: https://common-name-of-unit/.

16.9 POWER STRIP CONTROL

The console server supports a growing list of remote power-control devices (RPCs) which can be configured using the Management Console as described in Chapter 9. These RPCs are controlled using the open source PowerMan and Network UPS Tools and with Black Box's pmpower utility.

16.9.1 THE POWERMAN TOOL

PowerMan provides power management in a data center or compute cluster environment. It performs operations such as power on, power off, and power cycle via remote power controller (RPC) devices.

The powerman man page is not shipped with Black Box hardware. It is reproduced below.

Synopsis

powerman | pm [-options][targets]

TABLE 16-9. POWERMAN OPTIONS

OPTION	NOTES ABOUT TARGETS
-1 --on	power on targets
-0 --off	power off targets
-c --cycle	Power cycle targets
-r --reset	Assert hardware reset for targets (if implemented by RPC)
-f --flash	Turn beacon on for targets (if implemented by RPC)
-u --unflash	Turn beacon off for targets (if implemented by RPC)
-l --list	List available targets. If possible, output will be compressed into a host range (see target specification below)
-q --query	Query plug status of targets. If none specified, query all targets. Status is not cached; each time this option is used, powerman queries the appropriate RPC's. Targets connected to RPC's that could not be contacted (e.g. due to network failure) are reported as status "unknown". If possible, output will be compressed into host ranges.
-n --node	Query node power status of targets (if implemented by RPC). If no targets specified, query all targets. In this context, a node in the off state could be on at the plug but operating in standby power mode.
-b --beacon	Query beacon status (if implemented by RPC). If no targets are specified, query all targets.
-t --temp	Query node temperature (if implemented by RPC). If no targets are specified, query all targets. Temperature information is not interpreted by powerman and is reported as received from the RPC on one line per target, prefixed by target name.
-h --help	Display option summary.
-L --license	Show powerman license information
-d --destination	host[:port]. Connect to a powerman daemon on non-default host and optionally port.
-V --version	Display the powerman version number and exit.
-D --device	Displays RPC status information. If targets are specified, only RPC's matching the target list is displayed.
-T --telemetry	Causes RPC telemetry information to be displayed as commands are processed. Useful for debugging device scripts.
-x --exprange	Expand host ranges in query responses.

For more details see http://linux.die.net/man/1/powerman.

Target specification

powerman target hostnames may be specified as comma separated or space separated hostnames or host ranges.

Host ranges are of the general form:

prefix[n-m,l-k,...]

where n < m and l < k, etc.

This form should not be confused with regular expression character classes, which are also denoted by []. For example, foo[19] does not represent foo1 or foo9, but rather represents a degenerate range: foo19.

This range syntax is meant only as a convenience on clusters with a prefix NN naming convention and specification of ranges should not be considered necessary -- the list foo1,foo9 could be specified as such, or by the range foo[1,9].

Some examples of powerman targets follow.

Power on hosts bar,baz,foo01,foo02,...,foo05: powerman --on bar baz foo[01-05]

Power on hosts bar,foo7,foo9,foo10: powerman --on bar,foo[7,9-10]

Power on foo0,foo4,foo5: powerman --on foo[0,4-5]

As a reminder to the reader, some shells will interpret brackets — [and] — for pattern matching. Depending on your shell, it may be necessary to enclose ranged lists within quotes. For example, in tcsh, the last example above should be executed as:

```
powerman --on "foo[0,4-5]"
```

16.9.2 THE PMPOWER TOOL

The pmpower utility is a high level tool for manipulating remote preconfigured power devices connected to the console server either via a serial or network connection. The PDU UPS and IPMI power devices are variously controlled using the open source PowerMan, IPMItool or Network UPS Tools and Black Box's pmpower utility arches over these tools so the devices can be controlled through the one command line:

Synopsis

```
pmpower [-?h] [-l device | -r host] [-o outlet] [-u username]\
 [-p password] action
```

TABLE 16-10. PMPOWER OPTIONS

OPTION	NOTES
-? -h	This help message
-/	The serial port to use
-o	The outlet on the power target to apply to
-r	The remote host address for the power target
-u	Override the configured username
-p	Override the configured password
on	This action switches the specified device or outlet(s) on
off	This action switches the specified device or outlet(s) off
cycle	This action switches the specified device or outlet(s) off and on again
status	This action retrieves the current status of the device or outlet

Examples:

To turn outlet 4 of the power device connected to serial port 2 on:

```
# pmpower -l port02 -o 4 on
```

To turn an IPMI device off located at IP address 192.168.1.100 where the username is root and the password is calvin:

```
# pmpower -r 192.168.1.100 -u root -p calvin off
```

Default system Power Device actions are specified in /etc/powerstrips.xml.

Custom Power Devices can be added in /etc/config/powerstrips.xml. If an action is attempted which has not been configured for a specific Power Device pmpower will exit with an error.

16.9.3 ADDING NEW RPC DEVICES

There are a number of simple paths to adding support for new RPC devices.

The first is to have scripts to support the particular RPC included in either the open source PowerMan project—https://code.google.com/archive/p/powerman/—or the open source NUT UPS Tools project—http://networkupstools.org/.

The PowerMan device specifications are rather weird and it is suggested that you leave the actual writing of these scripts to the PowerMan authors. However documentation on how they work can be found at http://linux.die.net/man/5/powerman.dev. The Network UPS Tools (NUT) project has moved on from its UPS management origins to also cover SNMP PDUs (and embrace PowerMan). Black Box progressively includes the updated PowerMan and NUT build into the console server firmware releases.

The second path is to directly add support for the new RPC devices (or to customize the existing RPC device support) on your particular console server. The Manage > Power page uses information contained in /etc/powerstrips.xml to configure and control devices attached to a serial port. The configuration also looks for (and loads) /etc/config/powerstrips.xml if it exists.

You can add support for more devices by putting definitions for them into /etc/config/powerstrips.xml. This file can be created on a host system and copied to the Management Console device using scp. Alternatively, login to the Management Console and use ftp or wget to transfer files.

Here is a brief description of the elements of the XML entries in /etc/config/powerstrips.xml.

```
<powerstrip>
        <id>Name or ID of the device support</id>
        <outlet port="port-id-1">Display Port 1 in menu</outlet>
        <outlet port="port-id-2">Display Port 2 in menu</outlet>
        ...
        <on>script to turn power on</on>
        <off>script to power off</off>
        <cycle>script to cycle power</cycle>
        <status>script to write power status to
        /var/run/power-status</status>
        <speed>baud rate</speed>
        <charsize>character size</charsize>
        <stop>stop bits</stop>
        <parity>parity setting</parity>
</powerstrip>
```

The id appears on the web page in the list of available devices types to configure.

The outlets describe targets that the scripts can control. For example a power control board may control several different outlets. The port-id is the native name for identifying the outlet. This value will be passed to the scripts in the environment variable outlet, allowing the script to address the correct outlet.

There are four possible scripts: on, off, cycle and status.

When a script is run, it's standard input and output is redirected to the appropriate serial port. The script receives the outlet and port in the outlet and port environment variables respectively.

The script can be anything that can be executed within the shell.

All of the existing scripts in /etc/powerstrips.xml use the pmchat utility.

pmchat works just like the standard unix chat program, only it ensures interoperation with the port manager.

The final options, speed, charsize, stop and parity define the recommended or default settings for the attached device.

16.10 IPMITOOL

The console server includes the ipmitool utility for managing and configuring devices that support the Intelligent Platform Management Interface (IPMI) versions 1.5 and 2.0.

IPMI is an open standard for monitoring, logging, recovery, inventory, and control of hardware that is implemented independent of the main CPU, BIOS, and OS. The service processor (or Baseboard Management Controller, BMC) is the brain behind platform management and its primary purpose is to handle the autonomous sensor monitoring and event logging features.

The ipmitool program provides a simple command-line interface to this BMC. It features the ability to read sensor data repository (SDR) and print sensor values, display the contents of the System Event Log (SEL), print Field Replaceable Unit (FRU) inventory information, read and set LAN configuration parameters, and perform remote chassis power control.

The ipmitools man page is not shipped with Black Box hardware. It is reproduced below.

Synopsis

```
ipmitool [-c|-h|-v|-V] -I open <command>
ipmitool [-c|-h|-v|-V] -I lan -H <hostname>
        [-p <port>]
        [-U <username>]
        [-A <authtype>]
        [-L <privlvl>]
        [-a|-E|-P|-f <password>]
        [-o <oemtype>]
        <command>
ipmitool [-c|-h|-v|-V] -I lanplus -H <hostname>
        [-p <port>]
        [-U <username>]
        [-L <privlvl>]
        [-a|-E|-P|-f <password>]
        [-o <oemtype>]
        [-C <ciphersuite>]
        <command>
```

Description

This program lets you manage Intelligent Platform Management Interface (IPMI) functions of either the local system, via a kernel device driver, or a remote system, using IPMI V1.5 and IPMI v2.0. These functions include printing FRU information, LAN configuration, sensor readings, and remote chassis power control.

IPMI management of a local system interface requires a compatible IPMI kernel driver to be installed and configured. On Linux this driver is called OpenIPMI and it is included in standard distributions. On Solaris this driver is called BMC and is included in Solaris 10. Management of a remote station requires the IPMI-over-LAN interface to be enabled and configured. Depending on the particular requirements of each system it may be possible to enable the LAN interface using ipmitool over the system interface.

Options

TABLE 16-11. IPMTOOL OPTIONS

OPTION	VARIABLE	NOTES
-a		Prompt for the remote server password
-A	<authtype>	Present output in CSV (comma separated variable) format. This is not available with all commands.
-c		
-C	<ciphersuite>	The remote server authentication, integrity, and encryption algorithms to use for IPMIv2 lanplus connections. See table 22-19 in the IPMIv2 specification. The default is 3 which specifies RAKP-HMAC-SHA1 authentication, HMAC-SHA1-96 integrity, and AES-CBC-128 encryption algorithms.
-E		The remote server password is specified by the environment variable IPMI_PASSWORD.
-f	<password_file>	Specifies a file containing the remote server password. If this option is absent, or if password_file is empty, the password will default to NULL.
-h		Get basic usage help from the command line.
-H	<address>	Remote server address, can be IP address or hostname. This option is required for lan and lanplus interfaces.
-I	<interface>	Selects IPMI interface to use. Supported interfaces that are compiled in are visible in the usage help output.
-L	<privlvl>	Force session privilege level. Can be CALLBACK, USER, OPERATOR, and ADMIN. Default is ADMIN.
-m	<local_address>	Set the local IPMB address. The default is 0x20 and there should be no need to change it for normal operation.
-o	<oemtype>	Select OEM type to support. This usually involves minor hacks in place in the code to work around quirks in various BMCs from various manufacturers. Use -o list to see a list of current supported OEM types.
-p	<port>	Remote server UDP port to connect to. Default is 623.
-P	<password>	Remote server password is specified on the command line. If supported it will be obscured in the process list. Note! Specifying the password as a command line option is not recommended.
-t	<target_address>	Bridge IPMI requests to the remote target address.
-U	<username>	Remote server username, default is NULL user.
-v		Increase verbose output level. This option may be specified multiple times to increase the level of debug output. If given three times you will get hexdumps of all incoming and outgoing packets
-V		Display version information.

If no password method is specified then ipmitool will prompt the user for a password. If no password is entered at the prompt, the remote server password will default to NULL.

Security

The ipmitool documentation highlights that there are several security issues to be considered before enabling the IPMI LAN interface. A remote station has the ability to control a system's power state as well as being able to gather certain platform information. To reduce vulnerability it is strongly advised that the IPMI LAN interface only be enabled in 'trusted' environments where system security is not an issue or where there is a dedicated secure 'management network' or access has been provided through an console server.

Further it is strongly advised to not enable IPMI for remote access without setting a password. That that password should not be the same as any other password on that system.

When an IPMI password is changed on a remote machine with the IPMIv1.5 lan interface the new password is sent across the network as clear text. This could be observed and then used to attack the remote system. It is thus recommended that IPMI password management only be done over IPMIv2.0 lanplus interface or the system interface on the local station.

For IPMI v1.5, the maximum password length is 16 characters. Longer passwords are truncated. For IPMI v2.0, the maximum password length is 20 characters. Longer passwords are truncated.

Commands

TABLE 16-12. IPMI COMMANDS

COMMAND	NOTES
help	This can be used to get command-line help on ipmitool commands. It may also be placed at the end of commands to get option usage help.
raw	Send a RAW IPMI request and print response
lan	Configure LAN Channels
chassis*	Get chassis status and set power state
event	Send pre-defined events to MC
mc	Management Controller status and global enables
sdr	Print Sensor Data Repository entries and readings
sensor	Print detailed sensor information
fru	Print built-in FRU and scan SDR for FRU locators
sel	Print System Event Log (SEL)
pef	Configure Platform Event Filtering (PEF)
sol	Configure IPMIv2.0 Serial-over-LAN
isol	Configure IPMIv1.5 Serial-over-LAN
user	Configure Management Controller users
channel	Configure Management Controller channels
session	Print session information
exec	Run list of commands from file
set	Set runtime variable for shell and exec

*chassis commands: status, power, identify, policy, restart_cause, poh, bootdev.

chassis power commands: status, on, off, cycle, reset, diag, soft.

More details on ipmitools are availalbe at the project site, https://sourceforge.net/projects/ipmitool/.

16.11 CUSTOM DEVELOPMENT KIT (CDK)

As detailed in this manual, customers can copy scripts, binaries and configuration files directly to the console server.

Black Box also freely provides a development kit which allows changes to be made to the software in console server firmware image. The customer can use the CDK to:

- generate a firmware image without certain programs, such as telnet, which may be banned by company policy.
- generate an image with new programs, such as custom Nagios plug-in binaries or company specific binary utilities.
- generate an image with custom defaults e.g. it may be required that the console server be configured to have a specific default serial port profile which is reverted to even in event of a factory reset
- place configuration files into the firmware image, which cannot then be modified.

  For example:

  # /bin/config –-set= tools

  updates the configuration files in /etc/config which are read/write, whereas the files in /etc are read only and cannot be modified.

The CDK essentially provides a snapshot of the Black Box build process (taken after the programs have been compiled and copied to a temporary directory, romfs) just before the compressed file systems are generated.

You can obtain a copy of the Black Box CDK for the particular appliance you are working with from ftp://ftp.Black Box.com/cdk. Further information is avalable at http://Black Box.com/faq284.html.

16.12 SCRIPTS FOR MANAGING SLAVES

When the console servers are cascaded, the Master is in control of the serial ports on the Slaves, and the Master's Management Console provides a consolidated view of the settings for its own and all the Slave's serial ports.

The Master does not provide a fully consolidated view. Status > Active Users only displays those users active on the Master's ports. You will need to write a custom bash script that parses the port logs if you want to find out who's logged in to cascaded serial ports from the master.

You will probably also want to enable remote or USB logging, as local logs only buffer 8K of data and don't persist between reboots.

This script would, for example, parse each port log file line by line.

Each time it sees LOGIN: username, it adds username to the list of connected users for that port. Each time it sees LOGOUT: username, it removes it from the list.

The list can then be nicely formatted and displayed. It's also possible to run this as a CGI script on the remote log server.

To enable log storage and connection logging:

- select Alerts & Logging > Port Log.
- configure log storage.
- select Serial & Network > Serial Port.
- Edit the serial port(s).
- Under Console Server, select Logging Level 1.
- click Apply.

NOTE: A useful tutorial on creating a bash script CGI is at http://yolinux.com/TUTORIALS/LinuxTutorialCgiShellScript.html.

Similarly, the Master maintains a view of the status of the slaves:

- select Status > Support Report.
- scroll down to Processes.

- look for /bin/ssh -MN -o ControlPath=/var/run/cascade/%h slavename.

These are the slaves that are connected.

NOTE: The end of the Slaves' names will be truncated, so the first 5 characters must be unique.

Alternatively, you can write a custom CGI script as described above. The currently connected Slaves can be determined by running ls /var/run/cascade and the configured slaves can be displayed by running config -g config.cascade.slaves.

16.13 SMS SERVER TOOLS

Firmware releases v3.1 and later include the SMS Server Tools software that provides an SMS Gateway that can send and receive short messages through GSM modems and mobile phones.

You can send short messages by simply storing text files into a special spool directory. The program monitors this directory and sends new files automatically. It also stores received short messages into another directory as text files. Binary messages (including Unicode text) are also supported, for example ring tone messages. It's also possible to send a WAP Push message to a WAP- or MMS-capable mobile phone.

The program can be run as an SMS daemon which can be started automatically when the operating system starts. High availability can be ensured by using multiple GSM devices (currently up to 64).

The program can run other external programs or scripts after events like reception of a new message, successful sending and also when the program detects a problem. These programs can inspect the related text files and perform automatic actions.

The SMS Server Tools software needs a GSM modem (or mobile phone) with SMS command set according to the European specifications

GSM 07.05 (=ETSI TS 300 585) and GSM 03.38 (=ETSI TS 100 900).

The AT command set is supported. Devices can be connected with serial port, infrared or USB.

For more information see http://smstools3.kekekasvi.com/

16.14 MULTICAST

By default, all Black Box console servers come with Multicasting enabled. Multicasting provides Black Box products with the ability to simultaneously transmit information from a single device to a select group of hosts.

With firmware releases v3.1 and later, multicasting can be disabled and re-enabled from the command line. To disable multicasting type:

ifconfig eth0 -multicast

To re-enable multicasting from the command line type:

ifconfig eth0 multicast

IPv6 may need to be restarted when toggling between multicast states.

16.15 BULK PROVISIONING

Black Box appliances include wizard scripts to facilitate configuration and deployment en masse. These wizards operate at the command line level, so knowledge of the Linux command line and shell scripting is useful, but not necessary—they aim to be user-friendly enough for remote hands to manage. This bulk provisioning feature is supported by firmware version 3.9.1 or later, and VCMS version 4.4.0 and later (optional).

Both the bulk provisioning of Black Box appliances and bulk enrollment of these appliances into Virtual Central Management System (VCMS) is supported. These features may be used separately or in conjunction.

Using this method, an Black Box appliance can be fully configured and enrolled into VCMS with minimal interaction, in under 5 minutes. The basic steps are:

- Configure an individual golden master appliance with the baseline configuration shared by all Black Box appliances. This may be a minimal configuration if the installs are quite diverse, or a complete configuration when dealing with replicated installs.
- Use make-template to turn the golden master's active configuration into a template configuration that may be applied to other appliances.
- Create an OPG backup of the templated golden master appliance.
- Restore this configuration to each target devices via the CLI, web UI or using a USB thumb drive.
- Login via the CLI to complete configuration using setup-wizard.
- (Optional) On VCMS, use enrollment-wizard to automatically place appliances under management. This may be local/routable appliances, or remote appliances that have automatically Call Home using callhome-wizard.

Steps 5 and 6 may be reversed for remote setup via Lighthouse.

16.16 ZERO TOUCH PROVISIONING

Zero Touch Provisioning (ZTP) was introduced with firmware release 3.15.1 to allow Black Box appliances to be provisioned during their initial boot from a DHCP server.

16.16.1 PREPARATION

These are typical steps for configuration over a trusted network:

- Configure a same-model Black Box device.
- Optionally use the Bulk Provisioning wizard scripts to remove any appliance-specific settings (that is, create a template configuration) and/or prepare the configuration for automated VCMS enrollment. See Section 16.15.
- Save the configuration as an Black Box backup (.opg) file under System > Configuration Backup in the web UI, or via config -e in the CLI.

  Alternatively, you can save the XML configuration as a file ending in .xml.
- Publish the .opg or .xml file on a fileserver that understands one of the HTTPS, HTTP, FTP or TFTP protocols.
- Configure your DHCP server to include a vendor specific option for Black Box devices. The option text should be a URL to the location of the .opg or .xml file. The option text should not exceed 250 characters in length. It must end in either .opg or .xml.
- Connect a new Black Box device (either at defaults from the factory, or config erased) to the network and apply power.

NOTE: It may take up to 5 minutes for the device to find the .opg or .xml file via DHCP, download, install the file and reboot itself.

16.16.2 EXAMPLE ISC DHCP SERVER CONFIGURATION

The following is an example ISC DHCP server configuration fragment for serving an .opg configuration image:

```
option space Black Box code width 1 length width 1;
option Black Box.config-url code 1 = text;
        class "Black Box-ztp" {
                match if option vendor-class-identifier ~~ "^Black Box/";
                vendor-option-space Black Box;
        o       ption Black Box.config-url "https://example.com/opg/$
        {class}.opg";
}
```

For other DHCP servers, please consult their documentation on specifying vendor specific option fields.

We use sub-option 1 to hold the URL text.

16.16.3 SETUP FOR AN UNTRUSTED LAN

If network security is a concern and a user can insert a trusted USB flash drive into the Black Box device during provisioning, then follow the steps listed next for deploying configuration in an untrusted network:

- Generate an X.509 certificate for the client. Place it and its private key file onto a USB flash drive (concatenated as a single file, client.pem).
- Set up a HTTPS server that restricts access to the .opg or .xml file for HTTPS connections providing the client certificate.
- Put a copy of the CA cert (that signed the HTTP server's certificate) onto the USB flash drive as well (ca-bundle.crt).
- Insert the USB flash drive into the Black Box device before attaching power or network.
- Continue with the steps above, but using only an https URL.

16.16.4 HOW IT WORKS

This section explains in detail how the Black Box device uses DHCP to obtain its initial configuration.

First, a Black Box console manager is either configured or unconfigured. ZTP needs it to be in an unconfigured state, which is only obtained in the following ways:

- Firmware programming at factory.
- Pressing the Config Erase button twice during operation.
- Selecting Config Erase under System > Administration in the web UI, and rebooting.
- Creating the file /etc/config/.init and then rebooting.

When an unconfigured Black Box device boots, it performs these steps to find a configuration:

- The console server transmits a DHCP DISCOVER request onto its primary Network Interface (WAN).

  This DHCP request carries a Vendor Class Identifier of the form Black Box/model-name (for example, Black Box/LES1203A-M) and its parameter request list will include option 43 (Vendor-Specific Information).

- On receipt of a DHCP OFFER, the device will use the information in the offer to assign an IPv4 address to its primary Network Interface, add a default route, and prepare its DNS resolver.
- If the offer also contained an option 43 with sub-option 1, the device interprets the sub-option as a whitespace-separated list of URLs to configuration files to try to restore.
- If an NTP server option was provided in the DHCP offer, the system clock is synchronized with the NTP server.
- The system now searches all attached USB storage devices for two optional certificate files. The first file is named ca-bundle.crt and the second one is whichever one of the following filenames is found first:

TABLE 16-13. CERT FILENAMES

FILENAME	NOTES
client-aabbccddeeff.pem	AABBCCDDEEFF is the MAC address of the primary network interface.
client-model.pem	model is the (vendor class) model name in lowercase.
client.pem	—

- If both files—ca-bundle.crt and client*.pem—are found, then secure mode is enabled for the next section.

Each URL in the list obtained from option 43 sub-option 1 is tried in sequence until one succeeds:

- the URL undergoes substring replacement from the following table:

TABLE 16-14. URL SUB-STRING

SUB-STRING	REPLACED BY	NOTES
${mac}	the 12-digit MAC address of the device	0013b600b669
${model}	the full model name, in lowercase	les1716a-r2
${class}	the firmware hardware class	les1716a-r2
${version}	the firmware version number	3.15.1

- The resulting URL must end in .opg or .xml (an optional ?query-string is permitted).

  If is doesn't, it is skipped and the next URL is tried.
- In secure mode, the URL must use the https scheme or it is skipped.
- Otherwise, the available schemes are: http, https, tftp, ftp, and ftps.
- The curl program is used to download the URL.
- In secure mode, the server's certificate must validate against the ca-bundle.crt.

  The (required) client.pem file is provided to authenticate the client to the server. See the curl documentation for the format of these files.
- The URL is downloaded.

  For .opg files, its header is checked to see if it is compatible with the current device.

  For .xml files, a parse check is made. If the check fails, the downloaded file is abandoned and the next URL is tried.
- The file is imported into the current configuration.

- The system checks to see if a hostname has been set in the config. If not, it is set to ${model}-${mac}.
- The system checks to see if it is still in an unconfigured state. If it is, then the network interface mode is set to DHCP. This effectively forces the system into a configured state, preventing a future reboot loop.
- The system reboots.

NOTE: If all the URLs were skipped or failed, the system will wait for 30 seconds before retrying again. It will retry all the URLs up to 10 times. After the 10th retry, the system reboots. If the system has been manually configured in the meantime, the retries stop and ZTP is disabled.

NOTE: If no option 43 is received over DHCP, no URLs are downloaded and no reboots occur: the system must be manually configured. Once configured (manually or by ZTP), a console server will no longer request option 43 from the DHCP server, and it will ignore any option 43 configuration URLs presented to it.

16.17 INTERNAL STORAGE

Some models have an internal USB flash drive, a non-volatile NAND flash partition, or both, which can be used by portmanager for log storage and the TFTP/FTP server for file storage.

These storage devices are automatically mounted as subdirectories of /var/mnt/. The default directory served by FTP or TFTP is set to the preferred internal storage (if any), otherwise the first detected attached USB storage. The location of portmanager logs must be manually configured.

16.17.1 FILESYSTEM LOCATION OF FTP AND TFTP DIRECTORY

TABLE 16-15. FTP AND TFTP DIRECTORY

PRODUCT	PREFERRED STORAGE	DIRECTORY
LES1600	internal flash	/var/mnt/storage.nvlog/tftpboot/
LES1516A, LES1532A, LES1548A	internal USB flash	/var/mnt/storage.usb/tftpboot/
LES1700-R2	internal USB flash	/var/mnt/storage.usb/tftpboot/
Other products with USB	first-attached USB storage	/var/mnt/storage.usb/tftpboot/

16.17.2 FILESYSTEM LOCATION OF PORTMANAGER LOGS

TABLE 16-16. PORTMANAGER LOGS

PORT LOG SERVER TYPE	DIRECTORY
USB flash memory	/var/mnt/storage.usb
non-volatile internal storage	/var/mnt/storage.nvlog
micros-SD card	/var/mnt/storage.sd
other (NFS, CIFS, etc)	as explicitly configured

16.17.3 CONFIGURING FTP AND TFTP DIRECTORY

The FTP or TFTP services can be configured to serve different directories via the command line. For example:

```
config -s config.services.ftp.directory=/var/mnt/storage.usb/\
my-ftp-dir
config -r services
```

The directory will be created if it doesn't already exist.

16.17.4 MOUNTING A PREFERRED USB DISK BY LABEL

Currently, the "first" USB storage device is mounted at /var/mnt/storage.usb by detecting the lowest numbered disk partition, for example /dev/sda1. This can be constrained to match a particular port or a labelled device.

- Attach the USB disk you plan to use.
- Look in directories /dev/disk/by-path/ or /dev/disk/by-label/ to find a suitably stable way of identifying your disk.
- Use the following command to see the current device matching string used:
- # config -g config.storage.usb.device
- Change the path match with (for example):
- # config -s config.storage.usb.device=/dev/disk/by-label/1103

APPENDIX A: COMMANDS AND SOURCE CODE

A.1 COMMANDS

The console server platform is a dedicated Linux computer, optimized to provide monitoring and secure access to serial and network consoles of critical server systems and their supporting power and networking infrastructure.

Black Box console servers are built on the uCLinux distribution as developed by the uCLinux project. This is GPL code and the source can be found at http://uclinux.org/pub/uClinux/dist/.

Some uCLinux commands have config files that can be altered (for example, portmanager, inetd, init, and sshd).

Other commands you can run and do neat stuff with (for example loopback, bash (shell), ftp, hwclock, iproute, iptables, netcat, ifconfig, mii-tool, netstat, route, ping, portmap, pppd, routed, setserial, smtpclient, stty, stunel, tcpdump, tftp, tip, and traceroute).

Black Box console servers also ship with Busybox, the "Swiss Army Knife of embedded Linux" which "combines tiny versions of many common UNIX utilities into a single small executable." See https://busybox.net/ for more information.

The table below lists most of the standard uCLinux commands (ucl), Busybox commands (bb), and some custom Black Box commands (og), included in the default build tree. The shorthand immediately right of each listed command shows which source is used to run a given command: ucl for uCLinux; bb for Busybox; and og for Black Box-specific commands.

The Administrator can use these to configure the console server, and monitor and manage attached serial console and host devices.

TABLE A-1. COMMANDS

COMMAND		DESCRIPTION
addgroup	bb	Add a group or add a user to a group
adduser	bb	Add a user
agetty	ucl	Alternative Linux getty
arp	ucl	Manipulate the system ARP cache
arping	ucl	Send ARP requests/replies
bash	ucl	GNU Bourne-Again Shell
busybox	bb	Swiss army knife of embedded Linux commands
cat	bb	Concatenate file(s) and print them to stdout
chat	ucl	Useful for interacting with a modem connected to stdin/stdout
chgrp	bb	Change file access permissions
chmod	bb	Change file access permissions
chown	bb	Change file access permissions
config	og	Tool to manipulate and query system configuration from the shell
cp	bb	Copy files and directories
date	bb	Print or set the system date and time
dd	bb	Convert and copy a file.
deluser	bb	Delete a user from the system
df	bb	Report file system disk space usage
dhcpd	ucl	Dynamic Host Configuration Protocol server
discard	ucl	Network utility that listens on the discard port

TABLE A-1 (CONTINUED). COMMANDS

COMMAND		DESCRIPTION
dmesg	bb	Print or control the kernel ring buffer
echo	bb	Print the specified ARGs to stdout
erase	ucl	Tool for erasing MTD partitions
eraseall	ucl	Tool for erasing entire MTD partitions
false	bb	True and false return an exit status. Zero for true; non-zero for false
find	ucl	Search for files
flashw	ucl	Write data to individual flash devices
flatfsd	ucl	dæmon to save RAM file systems back to FLASH
ftp	ucl	Internet file transfer program
gen-keys	ucl	SSH key generation program
getopt	bb	Parses command options
gettyd	ucl	Getty dæmon
grep	bb	Print lines matching a pattern
gunzip	bb	Compress or expand files
gzip	bb	Compress or expand files
hd	ucl	ASCII, decimal, hexadecimal, octal dump
hostname	bb	Get or set hostname or DNS domain name
httpd	ucl	Listen for incoming HTTP requests
hwclock	ucl	Query and set hardware clock (RTC)
inetd	ucl	Network super-server dæmon
inetd-echo	ucl	Network echo utility
init	ucl	Process control initialization
ip	ucl	Show or manipulate routing, devices, policy routing, and tunnels
ipmitool	ucl	Linux IPMI manager
iptables	ucl	Administration tool for IPv4 packet filtering and NAT
ip6tables	ucl	Administration tool for IPv6 packet filtering
iptables-restore	ucl	Restore IP tables
iptables-save	ucl	Save IP tables
kill	bb	Send a signal to a process to end gracefully
ln	bb	Make links between files
login	ucl	Begin session on the system
loopback	og	Loopback diagnostic command
loopback1	og	Loopback diagnostic command
loopback2	og	Loopback diagnostic command
loopback8	og	Loopback diagnostic command

TABLE A-1 (CONTINUED). COMMANDS

COMMAND		DESCRIPTION
loopback16	og	Loopback diagnostic command
loopback48	og	Loopback diagnostic command
ls	bb	List directory contents
mail	ucl	Send and receive mail
mkdir	bb	Make directories
mkfs.jffs2	ucl	Create an MS-DOS file system under Linux
mknod	bb	Make block or character special files
more	bb	File persual filter for crt viewing
mount	bb	Mount a file system
msmtp	ucl	SMTP mail client
mv	bb	Move (rename) files
nc	ucl	TCP/IP Swiss army knife
netflash	ucl	Upgrade firmware on uCLinux platforms using the blkmem interface
netstat	ucl	Print network connections, routing tables, interface statistics, etc
ntpd	ucl	Network Time Protocol (NTP) dæmon
pgrep	ucl	Display process(es) selected by regex pattern
pidof	ucl	Find the process ID of a running program
ping	ucl	Send ICMP ECHO_REQUEST packets to network hosts
ping6	ucl	IPv6 ping
pkill	ucl	Sends a signal to process(es) selected by regex pattern
pmdeny	og	
pminetd	og	
pmloggerd	og	
pmshell	og	Similar to tip or cu but all serial port access is directed via portmanager.
portmanager	og	Command to handle all serial port access
portmap	ucl	DARPA port to RPC program number mapper
pppd	ucl	Point-to-point protocol dæmon
ps	bb	Report a snapshot of the current processes
pwd	bb	Print name of current working directory
reboot	bb	Soft reboot the system
rm	bb	Remove files or directories
rmdir	bb	Remove empty directories
routed	ucl	Show or manipulate the IP routing table
routef	ucl	IP route tool to flush IPv4 routes
routel	ucl	IP route tool to list routes

TABLE A-1 (CONTINUED). COMMANDS

COMMAND		DESCRIPTION
rtacct	ucl	network statistics tool
rtmon	ucl	RTnetlink listener
scp	ucl	Secure copy (remote file copy program)
sed	bb	Stream text editor
setmac	ucl	Sets the MAC address
setserial	ucl	Sets and reports serial port configuration
sh	ucl	The Bourne shell
showmac	ucl	Shows the MAC address
sleep	bb	Delay for a specified amount of time
smbmnt	ucl	Helper utility for mounting SMB file systems
smbmount	ucl	Mount an SMBFS file system
smbumount	ucl	SMBFS umount for normal users
snmpd	ucl	SNMP dæmon
snmptrap	ucl	Sends an SNMP notification to a manager
sredird	ucl	RFC2217-compliant serial port redirector
ssh	ucl	OpenSSH SSH client (remote login program)
ssh-keygen	ucl	Authentication kkey generation, management, and conversion
sshd	ucl	OpenSSH SSH dæmon
stty	ucl	Change and print terminal line settings
stunnel	ucl	Universal SSL tunnel
sync	bb	Flush file system buffers
sysctl	ucl	Configure kernel parameters at runtime
syslogd	ucl	System logging utility
tar	bb	The tar archiving utility
tc	ucl	Show traffic control settings
tcpdump	ucl	Dump traffic on a network
telnetd	ucl	Telnet protocol server
tftp	ucl	Client to transfer a file to or from a tftp server
tftpd	ucl	Trivial file transfer protocol (tftp) server
tip	ucl	Simple terminal emulator for connecting to modems and serial devices
top	ucl	Provide a view of process activity in real time
touch	bb	Change file timestamps
traceroute	ucl	Print the route packets take to a network host
traceroute6	ucl	Traceroute for IPv6
true	bb	True and false return an exit status. Zero for true; non-zero for false

TABLE A-1 (CONTINUED). COMMANDS

COMMAND		DESCRIPTION
umount	bb	Unmounts file systems
uname	bb	Print system information
usleep	bb	Delay for a specified time
vconfig	bb	Create and remove virtual ethernet devices
vi	bb	Busybox clone of the VI text editor
w	ucl	Show who is logged on and what they are doing.
zcat	bb	Identical to gunzip -c Bourne shell

With most of the above commands, the -h or --help argument provides a terse runtime description of their behavior.

More details on the Linux commands can found at http://en.tldp.org/HOWTO/HOWTO-INDEX/howtos.html and http://faqs.org/docs/Linux-HOWTO/Remote-Serial-Console-HOWTO.html.

Run ls when /bin/ is the present working directory (pwd) to view all the commands available on your console server.

There were a number of tools listed above, each denoted as og, that make it simple to configure the console server and ensure the changes are stored in the console server's flash memory. These commands are documented in previous chapters and include:

- config, which allows manipulation and querying of the system configuration from the command line. With config a new configuration can be activated by running the relevant configurator, which performs the action necessary to make the configuration changes live.
- portmanager, which provides a buffered interface to each serial port. It is supported by the pmchat and pmshell commands which ensure all serial port access is directed via the portmanager.
- pmpower, which is a configurable tool for manipulating remote power devices that are serially- or network-connected to the console server.
- SDT Connector, which is a java client applet that provides point-and-click SSH-tunneled connections to the console server and Managed Devices.

There are also a number of other CLI commands related to other open source tools embedded in the console server including:

- PowerMan provides power management for many preconfigured remote power controller (RPC) devices. For CLI details, see http://linux.die.net/man/1/powerman.
- Network UPS Tools (NUT) provides reliable monitoring of UPS and PDU hardware and ensure safe shutdowns of the systems that are connected, with a goal to monitor every kind of UPS and PDU. For CLI details see http://networkupstools.org/.
- Nagios is a popular, enterprise-class management tool that provides central monitoring of the hosts and services in distributed networks. For CLI details see http://nagios.org/.

The console server also supports GNU bash shell scripts, enabling the Administrator to run custom scripts. GNU bash, version 2.05.0(1)-release (arm-Black Box-linux-gnu) offers the following shell commands.

TABLE A-2. SHELL COMMANDS

COMMAND	ARGUMENTS
alias	[-p] [name[=value] …]
bg	[jobspec …]
bind	[-lpvsPVS] [-m keymap] [-f fi break [n]
case	word in [[(] pattern [| pattern]…) command-list ;;]… esac
cd	[-L|[-P [-e]] [-@] [directory]
command	[-pVv] command [arguments …]
compgen	[option] [word]
complete	[-abcdefgjksuv] [-o comp-option] [-DE] [-A action] [-G globpat] [-W wordlist] [-F function] [-C command] [-X filterpat] [-P prefix] [-S suffix] name [name …] complete -pr [-DE] [name …]
continue	[n]
declare	[-aAfFgilnrtux] [-p] [name[=value] …]
dirs	[-clpv] [+N | -N]
disown	[-ar] [-h] [jobspec … | pid …]
echo	[-neE] [arg …]
enable	[-a] [-dnps] [-f filename] [name …]
eval	[arguments]
exec	[-cl] [-a name] [command [arguments]]
exit	[n]
export	[-fn] [-p] [name[=value]]
false	
fc	[-e ename] [-lnr] [first] [last] -s [pat=rep] [command]
fg	[jobspec]
for	name [[in [words …]] ;] do commands; done
function	name { commands ; } or NA
getopts	optstring name [args]
hash	[-r] [-p filename] [-dt] [name]
help	[-dms] [pattern]
history	[n] -c -d offset [-anrw] [filename] -ps arg
if	test-commands; then consequent-commands; [elif more-test-commands; then more-consequents;] [else alternate-consequents;] fi
kill	[-s sigspec] [-n signum] [-sigspec] jobspec or pid -l|-L [exit_status]
local	[option] name[=value] …
logout	
popd	[-n] [+N | -N]

TABLE A-2 (CONTINUED). SHELL COMMANDS

COMMAND	ARGUMENTS
printf	[-v var] format [arguments]
pushd	[-n] [+N | -N | dir]
pwd	[-LP]
read	[-ers] [-a aname] [-d delim] [-i text] [-n nchars] [-N nchars] [-p prompt] [-t timeout] [-u fd] [name ...]
readonly	[-aAf] [-p] [name[=value]] ...
select	name [in words ...]; do commands; done
set	[--abefhkmnptuvxBCEHPT] [-o option-name] [argument ...] [+abefhkmnptuvxBCEHPT] [+o option-name] [argument ...]
shift	[n]
shopt	[-pqsu] [-o] [optname ...]
source	filename
suspend	[-f]
test	expr
time	[-lp]
times	
trap	[-lp] [arg] [sigspec ...]
true	
type	[-afptP] [name ...]
typeset	[-afFgrxilnrtux] [-p] [name[=value] ...]
umask	[-p] [-S] [mode]
unalias	[-a] [name ...]
unset	[-fnv] [name]
until	test-commands; do consequent-commands; done
variables	variable wait [n] while commands; do commands; done
while	test-commands; do consequent-commands; done

A.2 SOURCE CODE

Many console server software components are licensed under the GNU General Public License, Version 2. A copy of the GNU General Public License is included in Appendix 6: End-user license agreements. A copy is also available at http://gnu.org/licenses/old-licenses/gpl-2.0.html. Black Box will provide source code for any of the components of the software licensed under the GNU General Public License upon request.

The complete source code corresponding to each released version is available from us for a period of three years after its last shipment. If you would like the source code for an earlier release than the latest current release please write “source for firmware Version x.xx” in the memo line of your payment.

The console server also embodies the okvm console management software. This is GPL code and the full source is available from http://okvm.sourceforge.net/.

APPENDIX A: COMMANDS AND SOURCE CODE

The console server BIOS (boot loader code) is a port of uboot which is also a GPL package with source code openly available from http://denx.de/wiki/U-Boot/.

The console server CGIs (the html code, xml code and web config tools for the Management Console) are proprietary to Black Box. The code will be provided to customers, under NDA.

Also built-in to the console server is a Port Manager application and Configuration tools as documented in Chapters 15 and 16 above. These both are proprietary to Black Box, but open to customers under NDA, as above.

APPENDIX B: REGULATORY INFORMATION

B.1 FCC STATEMENT

This equipment has been found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

Shielded cables must be used with this equipment to maintain compliance with radio frequency energy emission regulations and ensure a suitably high level of immunity to electromagnetic disturbances.

All power supplies are certified to the relevant major international safety standards.

APPENDIX B: REGULATORY INFORMATION

B.2 NOM STATEMENT

1. Todas las instrucciones de seguridad y operación deberán ser leídas antes de que el aparato eléctrico sea operado.
2. Las instrucciones de seguridad y operación deberán ser guardadas para referencia futura.
3. Todas las advertencias en el aparato eléctrico y en sus instrucciones de operación deben ser respetadas.
4. Todas las instrucciones de operación y uso deben ser seguidas.
5. El aparato eléctrico no deberá ser usado cerca del agua—por ejemplo, cerca de la tina de baño, lavabo, sótano mojado o cerca de una alberca, etc.
6. El aparato eléctrico debe ser usado únicamente con carritos o pedestales que sean recomendados por el fabricante.
7. El aparato eléctrico debe ser montado a la pared o al techo sólo como sea recomendado por el fabricante.
8. Servicio—El usuario no debe intentar dar servicio al equipo eléctrico más allá a lo descrito en las instrucciones de operación. Todo otro servicio deberá ser referido a personal de servicio calificado.
9. El aparato eléctrico debe ser situado de tal manera que su posición no interfiera su uso. La colocación del aparato eléctrico sobre una cama, sofá, alfombra o superficie similar puede bloquea la ventilación, no se debe colocar en libreros o gabinetes que impidan el flujo de aire por los orificios de ventilación.
10. El equipo eléctrico deber ser situado fuera del alcance de fuentes de calor como radiadores, registros de calor, estufas u otros aparatos (incluyendo amplificadores) que producen calor.
11. El aparato eléctrico deberá ser connectado a una fuente de poder sólo del tipo descrito en el instructivo de operación, o como se indique en el aparato.
12. Precaución debe ser tomada de tal manera que la tierra fisica y la polarización del equipo no sea eliminada.
13. Los cables de la fuente de poder deben ser guiados de tal manera que no sean pisados ni pellizcados por objetos colocados sobre o contra ellos, poniendo particular atención a los contactos y receptáculos donde salen del aparato.
14. El equipo eléctrico debe ser limpiado únicamente de acuerdo a las recomendaciones del fabricante.
15. En caso de existir, una antena externa deberá ser localizada lejos de las lineas de energia.
16. El cable de corriente deberá ser desconectado del cuando el equipo no sea usado por un largo periodo de tiempo.
17. Cuidado debe ser tomado de tal manera que objectos liquidos no sean derramados sobre la cubierta u orificios de ventilación.
18. Servicio por personal calificado deberá ser provisto cuando:
    A: El cable de poder o el contacto ha sido dañado; u
    B: Objectos han caído o líquido ha sido derramado dentro del aparato; o
    C: El aparato ha sido expuesto a la lluvia; o
    D: El aparato parece no operar normalmente o muestra un cambio en su desempeño; o
    E: El aparato ha sido tirado o su cubierta ha sido dañada.

APPENDIX C: CONNECTIVITY, TCP PORTS AND SERIAL I/O

Pin-out standards exist for DB9 and DB25 connectors. There are, however, no pin-out standards for serial connectivity using RJ-45 connectors. Most console servers, serially-managed servers, routers, switches and power devices adopt their own unique pin-outs. Consequently, custom connectors and cables may be required to interconnect your console server.

C.1 SERIAL PORT PINOUTS

Console servers come with one to ninety-six serial connectors (notated SERIAL or SERIAL PORTS) for the RS-232 serial ports.

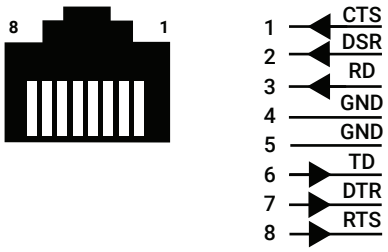
- The RJ-45 serial ports are located on the rear panel of the rack-mount LES1700-R2 series and LES1516A, LES1532A, LES1548A series).
- The LES1516A, LES1532A, LES1548A and LES1600 models have Cisco Straight serial pinouts on their RJ-45 connectors.
- The LES1700-R2 has software-selectable Cisco Straight or Cisco Rolled RJ-45.

CISCO STRAIGHT RJ-45 PINOUT

Straight through RJ-45 cable to equipment such as Cisco, Juniper, SUN, and more.

TABLE C-1. CISCO STRAIGHT RJ-45 PINOUT

PIN	SIGNAL	DEFINITION	DIRECTION
1	CTS	Clear to Send	Input
2	DSR	Data Set Ready	Input
3	RXD	Receive Data	Input
4	GND	Signal Ground	n/a
5	GND	Signal Ground	n/a
6	TXD	Transmit Data	Output
7	DTR	Data Terminal Ready	Output
8	RTS	Request to Send	Output

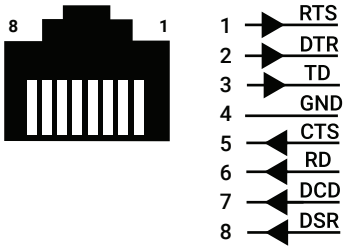


CISCO ROLLED RJ-45 PINOUT

Rolled RJ-45 cable to equipment such as Cisco, Juniper, SUN, and more.

TABLE C-2. CISCO ROLLED RJ-45 PINOUT

PIN	SIGNAL	DEFINITION	DIRECTION
1	RTS	Request to Send	Output
2	DTR	Data Terminal Ready	Output
3	TXD	Transmit Data	Output
4	GND	Signal Ground	n/a
5	CTS	Clear to Send	Input
6	RXD	Receive Data	Input
7	DCD	Data Carrier Detect	Input
8	DSR	Data Set Ready	Input



C.2 LOCAL CONSOLE PORT

Console servers with a dedicated LOCAL console/modem port use a standard DB9 connector for this port.

To connect to the LOCAL modem/console port on the console servers using a computer or terminal device using adapters with standard UTP CAT5 cable. Contact Black Box Technical Support at 877-877-2269 or info@blackbox.com for compatible adapters.

To connect the LOCAL console ports to modems (for out of band access) use an adapter with standard UTP CAT5 cable. Contact Black Box Technical Support at 877-877-2269 or info@blackbox.com for compatible adapters.

Each Black Box console server is supplied with UTP CAT5 cables.

C.3 RS-232 STANDARD PINOUTS

The RS-232 pinout standards for the DB9 and DB25 connectors are listed in the table below.

TABLE C-3. RS-232 STANDARD PINOUTS

PIN	SIGNAL	DB9	DEFINITION
1	—	–	Protective Ground
2	TXD	3	Transmitted Data
3	RXD	2	Received Data
4	RTS	7	Request to Send
5	CTS	8	Clear to Send
6	DSR	6	Data Set Ready
7	GND	5	Signal Ground
8	CD	1	Received line signal detector
9	—	–	Reserved for data set testing
10	—	–	Reserved for data set testing
11	—	—	Unassigned
12	SCF	–	Secondary received line signal detector
13	SCB	–	Secondary Clear to Send
14	SBA	–	Secondary Transmitted Data
15	DB	—	Transmission signal timing
16	SBB	–	Secondary Received Data
17	DD	–	Receiver signal element timing
18	—	–	Unassigned
19	SCA	—	Secondary Request to Send
20	DTR	4	Data Terminal Ready
21	CG	–	Signal quality detector
22	—	9	Ring Indicator
23	CH/CI	–	Data signal rate selector
24	DA	–	Transmit signal element timing
25	—	—	Unassigned

C.4 CONSOLE SERVER CONNECTOR WIRING

The LES1516A, LES1532A, LES1548A and LES1700-R2 families have the Cisco pinout by default and ship with cross-over/straight RJ-45-DB9 connectors.

DB9 TO RJ-45 STRAIGHT CONNECTOR

Straight through RJ-45 cable to equipment such as Cisco, Juniper, SUN, and more.

TABLE C-4. DB9F TO RJ-45S STRAIGHT CONNECTOR PINOUT

RJ-45 PIN	SIGNAL	WIRE	SIGNAL	DB9 PIN
1	CTS	----------	CTS	8
2	DCD	----------	DCD	1
3	RXD	----------	RXD	2
4	—	Not connected	—	—
5	GND	Signal Ground	GND	5
6	TXD	Transmit Data	TXD	3
7	DTR	Data Terminal Ready	DTR	4
8	RTS	Request to Send	RTS	7

C.5 TCP AND UDP PORT NUMBERS

Port numbers are divided into three ranges: Well Known Ports, Registered Ports and Dynamic & Private Ports. Well Known Ports are those from 0 through 1023. Registered Ports are those from 1024 through 49151. Dynamic & Private Ports are those from 49152 through 65535.

Well Known Ports are assigned by IANA, and on most systems, can only be used by system processes or by programs executed by privileged users. The table below shows some of the well-known port numbers. For more details, please visit the IANA website: http://www.iana.org/assignments/port-numbers.

TABLE C-5. TCP AND UDP PORT NUMBERS

PORT NUMBER	PROTOCOL	TCP/UDP
21	FTP (File Transfer Protocol)	TCP
22	SSH (Secure Shell)	TCP
23	Telnet	TCP
25	SMTP (Simple Mail Transfer Protocol)	TCP
37	Time	TCP, UDP
39	RLP (Resource Location Protocol)	UDP.
49	TACACS, TACACS+	UDP

TABLE C-5 (CONTINUED). TCP AND UDP PORT NUMBERS

PORT NUMBER	PROTOCOL	TCP/UDP
53	DNS	UDP
67	BootP server	UDP
68	BootP client	UDP
69	TFTP	UDP
70	Gopher	TCP
79	Finger	TCP.
80	HTTP	TCP
110	POP3	TCP
119	NNTP (Network News Transfer Protocol)	TCP
161/162	SNMP	UDP.
443	HTTPS	UDP

TABLE D-1. TERMINOLOGY

TERM	MEANING
3G	Third-generation cellular technology. The standards that determine 3G call for greater bandwidth and higher speeds for cellular networks.
AES	The Advanced Encryption Standard (AES) is a new block cipher standard to replace DES, developed by NIST, the US National Institute of Standards and Technology. AES ciphers use a 128-bit block and 128-, 192-, or 256-bit keys. The larger block size helps resist birthday attacks while the large key size prevents brute force attacks.
APN	Access Point Name (APN) is used by carriers to identify an IP packet data network that a mobile data user wants to communicate with and the type of wireless service.
Authentication	Authentication is the technique by which a process verifies that its communication partner is who it is supposed to be and not an imposter. Authentication confirms that data is sent to the intended recipient and assures the recipient that the data originated from the expected sender and has not been altered on route.
BIOS	Basic Input/Output System is the built-in software in a computer that are executed on startup (boot) and that determine what the computer can do without accessing programs from a disk. On PCs, the BIOS contains all the code required to control the keyboard, display screen, disk drives, serial communications, and a number of miscellaneous functions.
Bonding	Ethernet Bonding or Failover is the ability to detect communication failure transparently, and switch from one LAN connection to another.
BOOTP	Bootstrap Protocol. A protocol that allows a network user to automatically receive an IP address and have an operating system boot without user interaction. BOOTP is the basis for the more advanced DHCP.
Certificates	A digitally signed statement that contains information about an entity and the entity's public key, thus binding these two pieces of information together. A certificate is issued by a trusted organization (or entity) called a Certification Authority (CA) after the CA has verified that the entity is who it says it is.
Certificate Authority	A Certificate Authority is a trusted third party, which certifies public key's to truly belong to their claimed owners. It is a key part of any Public Key Infrastructure, since it allows users to trust that a given public key is the one they wish to use, either to send a private message to its owner or to verify the signature on a message sent by that owner.
Certificate Revocation List	A list of certificates that have been revoked by the CA before they expired. This may be necessary if the private key certificate has been compromised or if the holder of the certificate is to be denied the ability to establish a connection to the console server.
CHAP	Challenge-Handshake Authentication Protocol (CHAP) is used to verify a user's name and password for PPP Internet connections. It is more secure than PAP, the other main authentication protocol.
DES	The Data Encryption Standard is a block cipher with 64-bit blocks and a 56-bit key.
DHCP	Dynamic Host Configuration Protocol. A communications protocol that assigns IP addresses to computers when they are connected to the network.
DNS	The Domain Name System allocates Internet domain names and translates them into IP addresses. A domain name is a meaningful and easy to remember name for an IP address.
DUN	Dial-up Networking.
Encryption	The technique for converting a readable message (plaintext) into apparently random material (ciphertext) which cannot be read if intercepted. The proper decryption key is required to read the message.
Ethernet	A physical network layer protocol based upon IEEE standards.
Firewall	A network gateway device that protects a private network from users on other networks. A firewall is usually installed to allow users on an intranet access to the public Internet without allowing public Internet users access to the intranet.

TABLE D-1 (CONTINUED). TERMINOLOGY

TERM	MEANING
Gateway	A machine that provides a route (or pathway) to the outside world.
Hub	A network device that allows more than one computer to be connected as a LAN, usually using UTP cabling.
Internet	A worldwide system of computer networks - a public, cooperative, and self-sustaining network of networks accessible to hundreds of millions of people worldwide. The Internet is technically distinguished because it uses the TCP/IP set of protocols.
Intranet	A private TCP/IP network within an enterprise.
IP address	Fundamental internet addressing method that uses the form nnn.nnn.nnn.nnn.
IPMI	Intelligent Platform Management Interface (IPMI) is a set of common interfaces to a computer system which system administrators can use to monitor system health and manage the system. The IPMI standard defines the protocols for interfacing with a service processor embedded into a server platform.
Key lifetimes	The length of time before keys are re-negotiated.
LAN	Local Area Network.
LDAP	The Lightweight Directory Access Protocol (LDAP) is based on the X.500 standard, but significantly simpler and more readily adapted to meet custom needs. The core LDAP specifications are all defined in RFCs. LDAP is a protocol used to access information stored in an LDAP server.
LED	Light-Emitting Diode.
MAC address	Every piece of Ethernet hardware has a unique number assigned to it called its MAC address. Ethernet is used locally to connect the console server to the Internet, and it may share the local network with many other appliances. The MAC address is used by the local Internet router in order to direct console server traffic to it rather than something else in the local area. It is a 48-bit number usually written as a series of 6 hexadecimal octets. For example: 00:d0:cf:00:5b:da. A console server has a MAC address listed on a label underneath the device.
MSCHAP	Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is authentication for PPP connections between a computer using a Microsoft Windows operating system and a network access server. It is more secure than PAP or CHAP, and is the only option that also supports data encryption.
NAT	Network Address Translation. The translation of an IP address used on one network to an IP address on another network. Masquerading is one particular form of NAT.
Net mask	The way that computers know which part of a TCP/IP address refers to the network, and which part refers to the host range.
NFS	Network File System is a protocol that allows file sharing across a network. Users can view, store, and update files on a remote computer.
Out-of-band (OOB)	Out-of-Band (OOB) management is any management done over channels and interfaces that are separate from those used for user/customer data. Examples would include a serial console interface or a network interface connected to a dedicated management network that is not used to carry customer traffic, or to a BMC/service processor. Any management done over the same channels and interfaces used for user/customer data is In Band.
PAP	Password Authentication Protocol (PAP) is the usual method of user authentication used on the internet: sending a username and password to a server where they are compared with a table of authorized users. Whilst most common, PAP is the least secure of the authentication options.
PPP	Point-to-Point Protocol. A networking protocol for establishing simple links between two peers.

TABLE D-1 (CONTINUED). TERMINOLOGY

TERM	MEANING
RADIUS	The Remote Authentication Dial-In User Service (RADIUS) protocol was developed by Livingston Enterprises as an access server authentication and accounting protocol. The RADIUS server can support a variety of methods to authenticate a user. When it is provided with the username and original password given by the user, it can support PPP, PAP or CHAP, UNIX login, and other authentication mechanisms.
Router	A network device that moves packets of data. A router differs from a hub or a switch because it is intelligent and can route packets to their final destination.
SIM	Subscriber Identity Module (SIM) card stores unique serial numbers and security authentication used to identify a subscriber on mobile telephony devices.
SMASH	Systems Management Architecture for Server Hardware is a standards-based protocols aimed at increasing productivity of the management of a data center. The SMASH Command Line Protocol (SMASH CLP) specification provides an intuitive interface to heterogeneous servers independent of machine state, operating system or OS state, system topology or access method. It is a standard method for local and remote management of server hardware using out-of-band communication.
SMTP	Simple Mail Transfer Protocol. console server includes, SMTPclient, a minimal SMTP client that takes an email message body and passes it on to a SMTP server (default is the MTA on the local host).
SOL	Serial Over LAN (SOL) enables servers to transparently redirect the serial character stream from the baseboard universal asynchronous receiver/transmitter (UART) to and from the remote-client system over a LAN. With SOL support and BIOS redirection (to serial) remote managers can view the BIOS/POST output during power on, and reconfigured.
SSH	Secure Shell is secure transport protocol based on public-key cryptography.
SSL	ecure Sockets Layer is a protocol that provides authentication and encryption services between a web server and a web browser.
TACACS+	The Terminal Access Controller Access Control System (TACACS+) security protocol is a more recent protocol developed by Cisco. It provides detailed accounting information and flexible administrative control over the authentication and authorization processes. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide authentication, authorization, and accounting services independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon. There is a draft RFC detailing this protocol.
TCP/IP	Transmission Control Protocol/Internet Protocol. The basic protocol for Internet communication.
Telnet	Telnet is a terminal protocol that provides an easy-to-use method of creating terminal connections to a network.
UDP	User Datagram Protocol.
UTC	Co-ordinated Universal Time (equivalent to and replacement for GMT or Greenwich Mean Time).
UTP	Unshielded Twisted Pair cabling. A type of Ethernet cable that can operate up to 100Mb/s. Also known as Category 5 or CAT5.
VNC	Virtual Network Computing (VNC) is a desktop protocol to remotely control another computer. It transmits the keyboard presses and mouse clicks from one computer to another relaying the screen updates back in the other direction, over a network.
VPN	Virtual Private Network (VPN) a network that uses a public telecommunication infrastructure and Internet, to provide remote offices or individual users with secure access to their organization's network.
WAN	Wide Area Network.
WINS	Windows Internet Naming Service (WINS) that manages the association of workstation names and locations with IP addresses.

APPENDIX E: DISCLAIMER/TRADEMARKS

E.1 DISCLAIMER

Black Box Corporation shall not be liable for damages of any kind, including, but not limited to, punitive, consequential or cost of cover damages, resulting from any errors in the product information or specifications set forth in this document and Black Box Corporation may revise this document at any time without notice.

E.2 TRADEMARKS USED IN THIS MANUAL

Black Box and the Black Box logo type and mark are registered trademarks of Black Box Corporation.

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

NEED HELP?
LEAVE THE TECH TO US

LIVE 24/7 TECHNICAL SUPPORT

1.877.877.2269

BLACK BOX®