Owner's Manual

Console Server Management Switch

Models:
B096-016 / B096-032 / B096-048

Console Server with PowerAlert

Model:
B092-016

Console Server

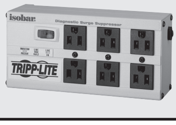
Models:
B095-004-1E / B095-003-1E-M / B094-008-2E-M-F / B094-008-2E-V

PROTECT YOUR INVESTMENT!

Register your product for quicker service and ultimate peace of mind.

You could also win an ISOBAR6ULTRA surge protector—a $50 value!

www.tripplite.com/warranty



TRIPP·LITE

OVER 90 YEARS
Manufacturing
Excellence.
Since 1922.

1111 W. 35th Street, Chicago, IL 60609 USA • www.tripplite.com/support

Copyright © 2014 Tripp Lite. All rights reserved. All trademarks are the property of their respective owners.

FCC Information, Class A

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and 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. The user must use shielded cables and connectors with this equipment. Any changes or modifications to this equipment not expressly approved by Tripp Lite could void the user's authority to operate this equipment.

RoHS

This product is RoHS compliant.

User Notice

All information, documentation and specifications contained in this manual are subject to change without prior notification by the manufacturer. The manufacturer makes no representations or warranties, either expressed or implied, with respect to the contents hereof and specifically disclaims any warranties as to merchantability or fitness for any particular purpose. Any of the manufacturer's software described in this manual is sold or licensed `as is'. Should the programs prove defective following their purchase, the buyer (and not the manufacturer, its distributor, or its dealer), assumes the entire cost of all necessary servicing, repair and any incidental or consequential damages resulting from any defect in the software.
The manufacturer of this system is not responsible for any radio and/or TV interference caused by unauthorized modifications to this device. It is the responsibility of the user to correct such interference. The manufacturer is not responsible for any damage incurred in the operation of this system if the correct operational voltage setting was not selected prior to operation.

⚠ Please take care to 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 which may cause fire or electric shock. Refer all service to Tripp Lite 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
- Also it is recommended you use a surge suppressor or UPS to protect the equipment from transients

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Chapter 1: Introduction

This User Manual is provided to help you get the most from your B096-016 / B096-032 / B096-048 Console Server Management Switch, B092-016 Console Server with PowerAlert or B095-004-1E / B095-003-1E-M / B094-008-2E-M-F / B094-008-2E-V Console Server product. These products are referred to generically in this manual as Console Servers.

Once configured, you will be able to use your Console Server to securely monitor, access and control the computers, networking devices, telecommunications equipment, power supplies and operating environment in your data center, branch office or communications room. This manual guides you in managing this infrastructure locally (at the rack side or across your operations or management LAN or through the local serial console port), and remotely (across the Internet, private network or via dial up).

Manual Organization

This manual contains the following chapters:

1. Introduction	An overview of the features of the Console Server and information on this manual
2. Installation	Details physical installation of the Console Server and the interconnection of controlled devices
3. System Configuration	Describes the initial installation and configuration using the Management Console of the Console Server on the network and the services that will be supported
4. Serial and Network	Covers configuring serial ports and connected network hosts, and setting up Users and Groups
5. Failover and OoB dial-in	Describes setting up the high-availability access features of the Console Server
6. Secure Tunneling (SDT)	Covers secure remote access using SSH and configuring for RDP, VNC, HTTP, HTTPS, etc. access to network and serially connected devices
7. Alerts and Logging	Explains the setting up of local and remote event/ data logs and triggering SNMP and email alerts
8. Power & Environment	Management of USB, serial and network attached Power Distribution units and UPS units including Network UPS Tool (NUT) operation and IPMI power control. EMD environmental sensor configuration
9. Authentication	All access to the Console Server requires usernames and passwords which are locally or externally authenticated
10. Nagios Integration	Setting Nagios central management with SDT extensions and configuring the Console Server as a distributed Nagios server
11. System Management	Covers access to and configuration of services to be run on the Console Server
12. Status Reports	View the status and logs of serial and network connected devices (ports, hosts, power and environment)
13. Management	Includes port controls and reports that can accessed by Users
14. Basic Configuration	Command line installation and configuration using the config command
15. Advanced Config	More advanced command line configuration activities where you will need to use Linux commands
16. Thin Client	Configuration and use of the thin client and other applications (including PowerAlert) embedded in the Console Server with PowerAlert (B092-016) product

Types of users

The Console Server supports two classes of users:

I. Administrative users: Those who will be authorized to configure and control the Console Server; and to access and control all the connected devices. These administrative users will be set up as members of the admin user group. Any user in this class is referred to generically in this manual as an Administrator. 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).

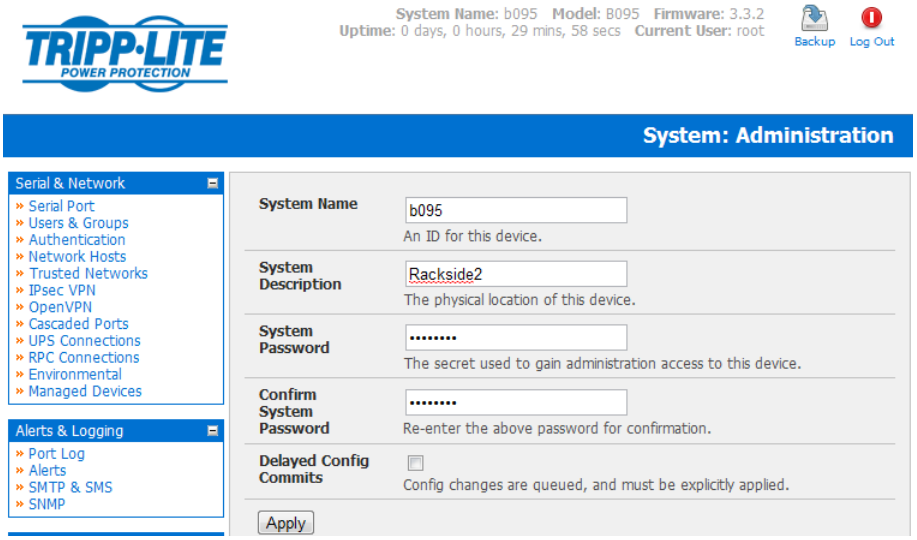
II. Users: Embraces those who have been set up by the Administrator with specific limits on their access and control authority. These users are set up as members of the user's user group (or some other user groups the Administrator may have added). They are only authorized to perform specified controls 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 (e.g. Telnet, HHTPS, RDP, IPMI, Serial over LAN, Power Control). An authorized User can also use the Management Console to access configured devices and review port logs.

In this manual, when the term user (lower case) is used, it is referring to both the above classes of users. 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, or it 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 Console Server Management Console runs in a browser. It provides a view of your Console Server Management Switch (B096-016/032/048), Console Server with PowerAlert (B092-016) or Console Server (B095-004/003 and B094-008-2E-M-F / B094-008-2E-V) product and all the connected equipment.

Administrators can use the Management Console, either locally or from a remote location, to configure the Console Server, set up Users, configure the ports and connected hosts, and set up logging and alerts.



System Name: b095 Model: B095 Firmware: 3.3.2
Uptime: 0 days, 0 hours, 29 mins, 58 secs Current User: root

System: Administration

System Name: b095
An ID for this device.

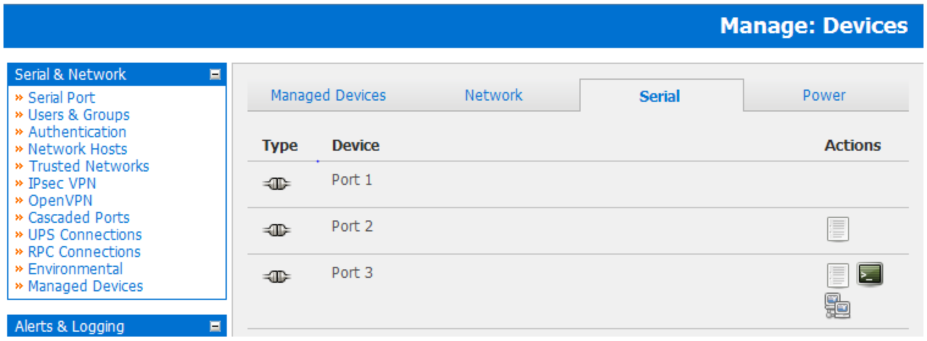
System Description: Rackside2
The physical location of this device.

System Password: ••••••••
The secret used to gain administration access to this device.

Confirm System Password: ••••••••
Re-enter the above password for confirmation.

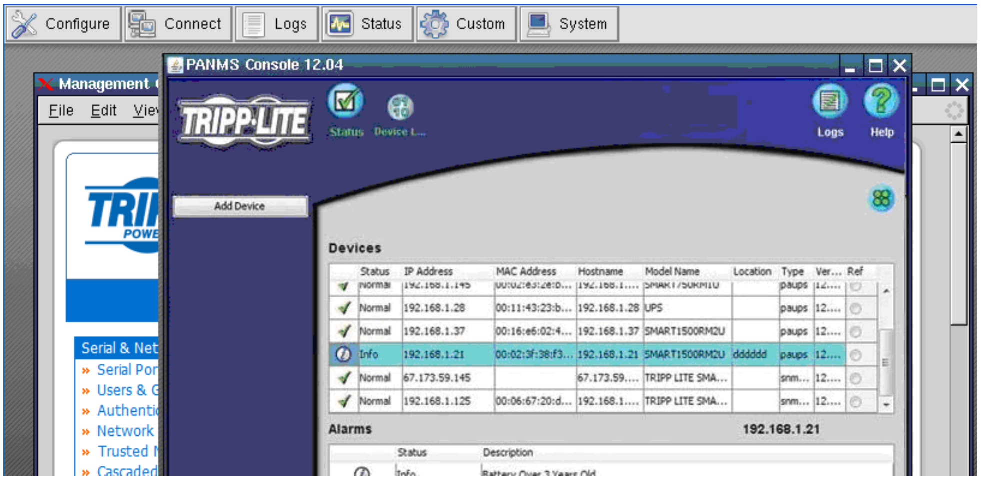
Delayed Config Commits: □
Config changes are queued, and must be explicitly applied.

An authorized User can use the Management Console to access and control configured devices, review port logs, use the in-built Web terminal to access serially attached consoles and control power to connected devices.



Type	Device	Actions
	Port 1	
	Port 2	
	Port 3	

The Console Server runs an embedded Linux operating system. Experienced Linux and UNIX users may prefer to undertake configuration at the command line. As an Administrator you can get command line access by connecting through a terminal emulator or communications program to the console serial port; or by SSH or Telnet connecting to the Console Server over the LAN; or by connecting to the Console Server through an SSH tunnel using the SDTConnector.
The B092-016 Console Server also has PowerAlert software and a selection of thin clients embedded (RDP, Firefox etc). You will be able to use these consoles as well as the standard Management Console for access and control.



Devices

	Status	IP Address	MAC Address	Hostname	Model Name	Location	Type	Ver...	Ref
✓	Normal	192.168.1.145	00:02:e3:2e:b...	192.168.1....	SMART750RM1U		paups	12....	○
✓	Normal	192.168.1.28	00:11:43:23:b...	192.168.1.28	UPS		paups	12....	○
✓	Normal	192.168.1.37	00:16:e6:02:4...	192.168.1.37	SMART1500RM2U		paups	12....	○
ⓘ	Info	192.168.1.21	00:02:3f:38:f3...	192.168.1.21	SMART1500RM2U	dddddd	paups	12....	○
✓	Normal	67.173.59.145		67.173.59....	TRIPP LITE SMA...		snm...	12....	○
✓	Normal	192.168.1.125	00:06:67:20:d...	192.168.1....	TRIPP LITE SMA...		snm...	12....	○

Alarms 192.168.1.21

	Status	Description
ⓘ	Info	Battery Over 3 Years Old

Manual Conventions

This manual uses different fonts and typefaces to show specific actions:
Note: *Text presented like this indicates issues to take note of.*

> ⚠ Text presented like this highlights important issues and it is essential you read and take head of these warnings.

- Text presented with a bullet point indent indicates an action you should take as part of the procedure.

Bold text indicates text that you type, or the name of a screen object (e.g. a menu or button) on the Management Console.

Italic text is also used to indicate a text command to be entered at the command line level.

Publishing history

Date	Revision	Update details
January 2009	0.9	Initial draft
February 2009	0.91	Pre-release
January 2010	1.01	Add B095-004/003 Console Server and Firmware 3.0.1 features
January 2011	2.0	Firmware 3.3.2 features
March 2011	2.0.1	Support for additional USB ports and 16GB internal flash in B096-016 / B096-032 / B096-048
February 2012	2.0.02	Add B094-008-2E-M-F and 3.5.2 firmware features
September 2013	2.0.3	Firmware 3.8.1 features
October 2014	2.0.4	Add B094-008-2E-V and 3.11.2 firmware features

Chapter 2: Installation

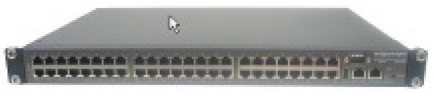
This chapter describes the physical installation of the Console Server hardware and connection to controlled devices

2.1 Models

There are a number of Console Server models, each with a different number of network, USB and serial ports and power supplies:

Console Server Model	Serial Ports	Network Ports	Console Port	USB Port	Modem	Power
B096-048	48	2	1	1+2	Internal	Dual AC Universal Input
B096-032	32	2	1	1+2	Internal	Dual AC Universal Input
B096-016	16	2	1	1+2	Internal	Dual AC Universal Input
B092-016	16	1	1+KVM	4	-	Single AC Universal Input
B095-004-1E	4	1	1	1	-	External DC Supply
B095-003-1E-M	3	1	1	1	Internal	External DC Supply
B094-008-2E-M-F	8	2	1	2	Internal	External DC Supply
B094-008-2E-V	8	2	1	2	Internal Cellular	External DC Supply

2.1.1 Kit components: B096-048, B096-032 and B096-016 Console Server Management Switch



B096-048, B096-032 or B096-016
Console Server Management Switch

2 x Cable UTP Cat5 blue

Connectors
DB9F-RJ45S straight and cross-over

Dual IEC AC power cords

Quick Start Guide and CD-ROM

- Unpack your Console Server Management Switch kit and verify you have all the parts shown above, and that they all appear in good working order
- If you are installing your Console Server Management Switch in a rack you will need to attach the rack mounting brackets supplied with the unit, and install the unit in the rack. Take care to head the Safety Precautions
- Connect your Console Server Management Switch to the network, to the serial ports of the controlled devices, and to power as outlined below

2.1.2 Kit components: B092-016 Console Server with PowerAlert

B092-016
Console Server with PowerAlert

2 x Cable UTP Cat5 blue

Connector
DB9F-RJ45S straight and DB9F-RJ45S cross-over

AC power cable

Quick Start Guide and CD-ROM

- Unpack your Console Server and verify you have all the parts shown above, and that they all appear in good working order
- If you are installing your Console Server in a rack, you will need to attach the rack mounting brackets supplied with the unit, and install the unit in the rack. Take care to heed the Safety Precautions listed earlier
- Proceed to connect your B092-016 to the network, to the serial and USB ports of the controlled devices, to any rack side LCD console or KVM switch, and to power as outlined below

2.1.3 Kit components: B095-004-1E and B095-003-1E-M Console Server

B095-004-1E 4-port Console Server with single NIC or B095-003-1E-M 3- port Console Server with single NIC and modem

2 x Cable UTP Cat5 blue

Connectors
DB9F-RJ45S straight and cross-over

External power supply

Quick Start Guide and CD-ROM

- Unpack your Console Server kit and verify you have all the parts shown above, and that they all appear in good working order
- If you are installing your Console Server in a rack you will need to attach the rack mounting brackets supplied with the unit, and install the unit in the rack. Take care to head the Safety Precautions
- Proceed to connect your Console Server to the network, to the serial ports of the controlled devices, and to power as outlined below

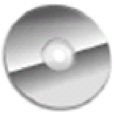
2.1.4 Kit components: B094-008-2E-M-F and B094-008-2E-V Console Server

B094-008-2E-M-F 8- port Console Server with dual NIC and modem or B094-008-2E-V 8 -port Console Server with dual NIC and cellular

2 x Cable UTP Cat5 blue

Connectors
DB9F-RJ45S straight and cross-over

External power supply



Quick Start Guide and CD-ROM

- Unpack your Console Server kit and verify you have all the parts shown above, and they all appear to be in good working order
- If you are installing your Console Server in a rack, you will need to attach the rack mounting brackets supplied with the unit and install the unit in the rack. Follow the Safety Precautions
- Proceed to connect your Console Server to the network, to the serial ports of the controlled devices, and to power as outlined below

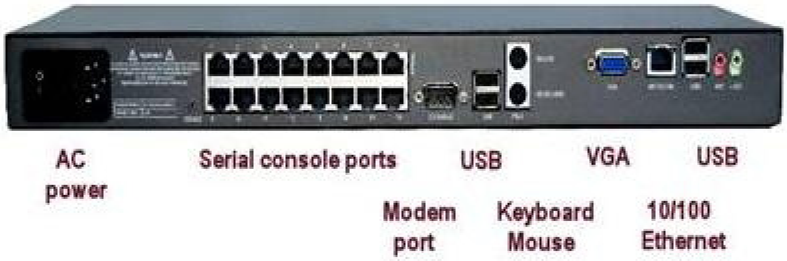
2.2 Power Connection

2.2.1 Power: Console Server Management Switch

The B096-048/032/016 Console Server Management Switch has dual universal AC power supplies with auto failover built in. These power supplies each 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 30W. 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. A North American power cord is provided by default. Power cords for other regions are available separately from Tripp Lite.

2.2.2 Power: Console Server with PowerAlert

The standard B092-016 Console Server has a built-in universal auto-switching AC power supply. This power supply accepts AC input voltage between 100 and 240 VAC with a frequency of 50 or 60 Hz and the power consumption is less than 40W.



The AC power socket is located at the rear of the B092-016. This power inlet uses a conventional AC power cord. A North American power cord is provided by default. Power cords for other regions are available separately from Tripp Lite.

2.2.3 Power: Console Server

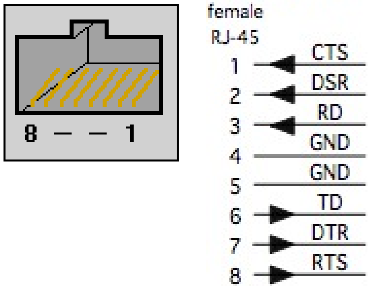
The B095-004/003 and B094-008-2E-M-F / B094-008-2E-V Console Servers each have an external wall-mount power supply. This power supply accepts AC input voltage between 100 and 240 VAC with a frequency of 50 or 60 Hz and the total power consumption per console server is less than 20W. The DC power socket on the Console Server is located on the side of the metal case marked PWR.

2.3 Network Connection

The RJ45 10/100 LAN port is located on the rear of the B092-016 Console Server, on the front of the B096-048/032/016 Console Server Management Switch and on the side panel of the B095-004/003 and B094-008-2E-M-F / B094-008-2E-V Console Servers. All physical connections are made using industry standard Cat5e patch cables (Tripp Lite N001 and N002 series cables). Ensure you only connect the LAN port to an Ethernet network that supports 10Base-T/100Base-T. For the initial configuration of the Console Server you must connect a computer to the Console Server's principal network port.

2.4 Serial Port Connection

The RJ45 serial ports are located on the rear of the B092-016 Console Server, on the front of the B096-048/032/016 Console Server and B094-008 Console Server, and on the side panel of the B095-004/003 Console Server. These Console Servers use the RJ45 pinout used by Cisco. Use straight through RJ-45 cabling to connect to equipment such as Cisco, Juniper, SUN, and more.



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	NA
5	GND	Signal Ground	NA
6	TXD	Transmit Data	Output
7	DTR	Data Terminal Ready	Output
8	RTS	Request To Send	Output

Conventional Cat5 cabling with RJ45 jacks are used for serial connections. Before connecting the console port of an external device to the Console Server serial port, confirm that the device supports standard RS-232C (EIA-232).

The Console Server also has a DB9 LOCAL (Console/Modem) port. This DB-9 connector is on the rear panel of the B092-016 Console Server, and on the front panel of the B096-048/032/016 Console Server Management Switch.

2.5 USB Port Connection

The B096-048/032/016 Console Server Management Switch has one USB 1.0 port on the front panel and two USB 2.0 ports on the rear. External USB devices can be plugged into these USB ports.

***Note**: The B096-048/032/016 Console Server Management Switch ships with an internal 16GB USB memory which can be used for extended log file storage*

The B094-008-2E-M-F / B094-008-2E-V Console Server has two USB 2.0 ports on the front. External USB devices can be plugged into these USB ports.

***Note**: The B094-008-2E-M-F / B094-008-2E-V Console Server ships with an internal 4GB USB memory which can be used for extended log file storage*

There are four USB 2.0 ports on the rear panel of the B092-016 Console Server and one USB2.0 port located under the RJ45 10/100 LAN connector on the B095-004/003 Console Server. These ports are used to connect to USB consoles (of managed UPS hardware) and to other external devices (such as a USB memory stick or keyboard).

External USB devices (including USB hubs) can be plugged into any Console Server USB port.

2.6 Rackmount Console / KVM Connection (B092-016 only)

B092-016 Console Server with PowerAlert can be connected directly to a rackmount console (such as B021-000-17 or B021-019 by Tripp Lite) to provide direct local management right at the rack. Connect the rackmount console's PS/2 Keyboard/Mouse and VGA connectors directly to the PS/2 and VGA connectors on the B092-016. The default video resolution is 1024 x 768. The B092-016 Console Server also supports the use of a USB keyboard/mouse.

Alternately, the B092-016 Console Server can also be connected locally to a KVM (or KVMoIP) switch at the rack. The B092-016 Console Server with PowerAlert will enable you then to use this KVM infrastructure to run PowerAlert, to manage your power devices and to run the thin clients to manage other devices.

***Note**: Care should be taken in handling all Console Server products. There are no operator-serviceable components inside, so do not remove cover. Refer any service to qualified personnel*

Chapter 3: Initial System Configuration

This chapter provides step-by-step instructions for the initial configuration of your Console Server and connecting it to your management or operational network. This involves the Administrator:

- Activating the Management Console
- Changing the Administrator password
- Setting the IP address for the Console Server's principal LAN port
- Selecting the network services to be supported

This chapter also discusses the communications software tools that the Administrator may use to access the Console Server. It also covers the configuration of the additional LAN ports on the B096-016/032/048 Console Server Management Switch.

3.1 Management Console Connection

Your Console Server comes configured with a default IP Address 192.168.0.1 Subnet Mask 255.255.255.0

- Directly connect a computer to the Console Server

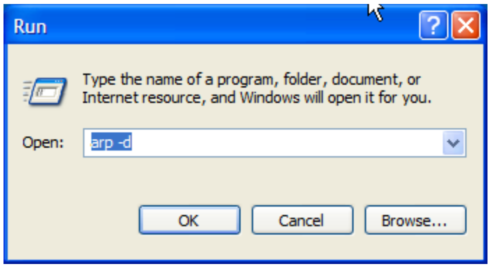
***Note**: For initial configuration it is recommended that the Console Server be connected directly to a single PC or computer. However, if you choose to connect your LAN before completing the initial setup steps, it is important that:*

- o *you ensure there are no other devices on the LAN with an **address of 192.168.0.1***
- o *the Console Server and the computer are on the same LAN segment, with no interposed router appliances*

3.1.1 Connected computer set up

To configure the Console Server with a browser, the connected computer 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 your Linux or Unix computer simply run ifconfig
- For Windows PCs (Win9x/Me/2000/XP/Vista/7/NT):
  - Click **Start** -> (**Settings** ->) **Control Panel** and double click **Network Connections** (for 95/98/Me, double click **Network**).
  - 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:
    - o IP address: **192.168.0.100**
    - o 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 as a secondary IP connection.
- If it is not convenient to change your computer network address, you can use the *ARP-Ping* command to reset the Console Server IP address. To do this from a Windows PC:
  - Click **Start** -> **Run** (or select **All Programs** then **Accessories** then **Run**).
  - Type *cmd* and click **OK** to bring up the command line.
  - Type *arp –d* to flush the ARP cache.
  - Type *arp –a* to view the current ARP cache (this should be empty).



Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

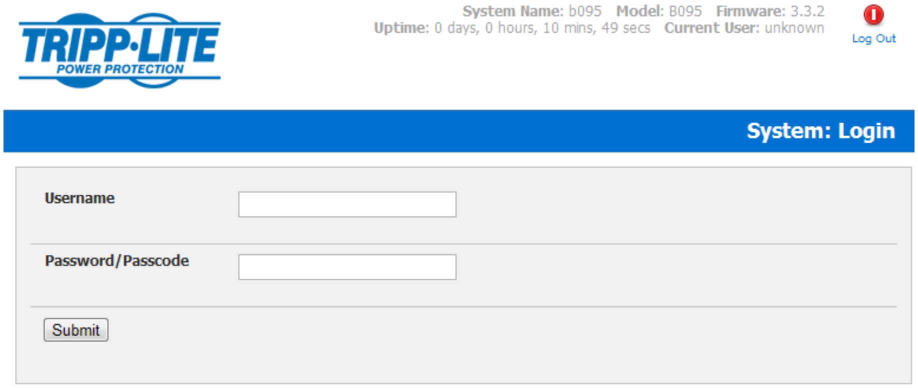
Open: arp -d

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 a MAC Address 00:13:C6:00:02:0F (designated on the label on the bottom of the unit) and we are setting its IP address to 192.168.100.23. Also the PC/workstation 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)

- Type *arp -s 192.168.100.23 00-13-C6-00-02-0F* (Note for UNIX the syntax is: *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.

3.1.2 Browser connection

- Activate your preferred browser on the connected computer and enter **https://192.168.0.1** The Management Console supports all current versions of the popular browsers (Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari and more)



System Name: b095 Model: B095 Firmware: 3.3.2
Uptime: 0 days, 0 hours, 10 mins, 49 secs Current User: unknown

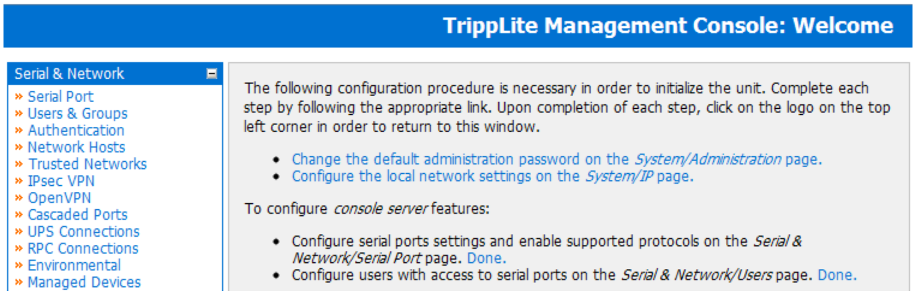
System: Login

Username

Password/Passcode

- You will be prompted to log in. Enter the default administration username and administration password (Username: **root** Password: **default**)

Note*: Console Servers are factory configured with HTTPS access enabled and HTTP access disabled.*



TrippLite Management Console: Welcome

The following configuration procedure is necessary in order to initialize the unit. Complete each step by following the appropriate link. Upon completion of each step, click on the logo on the top left corner in order to return to this window.

- Change the default administration password on the *System/Administration* page.
- Configure the local network settings on the *System/IP* page.

To configure *console server* features:

- Configure serial ports settings and enable supported protocols on the *Serial & Network/Serial Port* page. Done.
- Configure users with access to serial ports on the *Serial & Network/Users* page. Done.

A **Welcome** screen, which lists initial installation configuration steps, will be displayed. These steps are:

- Change default administration password (System/Administration page. Refer Chapter 3.2)
- Configure the local network settings (System/IP page. Refer Chapter 3.3)

To configure Console Server features:

- Configure serial ports settings (Serial & Network/Serial Port page. Refer Chapter 4)
- Configure user port access (Serial & Network/Users page. Refer Chapter 4)

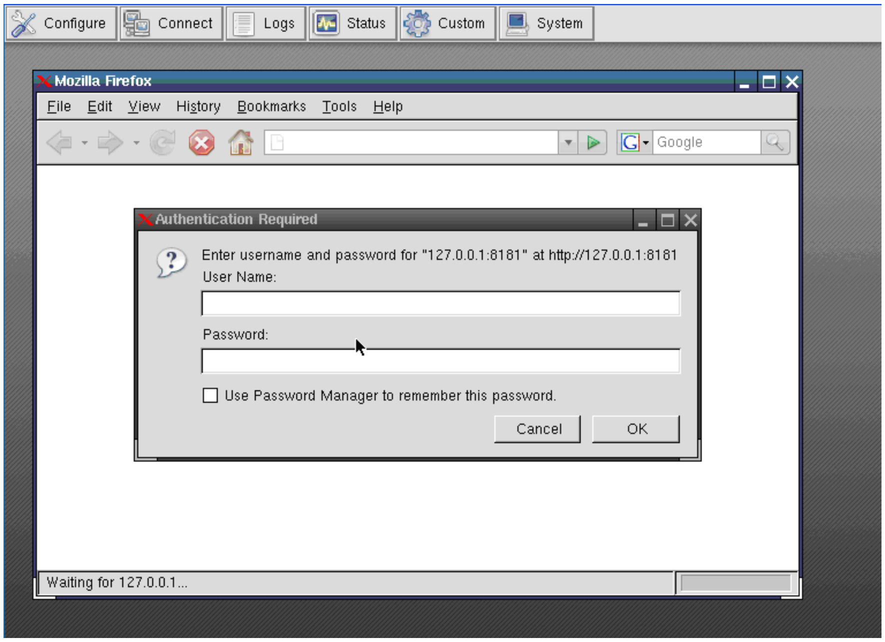
After completing each of the above steps, you can return to the configuration list by clicking the Tripp Lite logo in the top left corner of the screen:

***Note**: If you are not able to connect to the Management Console at 192.168.0.1 or if the default Username / Password were not accepted then reset your Console Server (refer Chapter 10)*

3.1.3 Initial B092-016 connection

You can configure the B092-016 Console Server using a connected computer and browser connection as described in the two sections above, or you can configure it directly. To do this you will need to connect a console (keyboard, mouse and display) or a KVM switch directly to its mouse, keyboard and VGA ports. When you initially power on the B092-016, you will be prompted on your directly connected video console to log in

- Enter the default administration username and password (Username: **root** Password: **default**). The B092-016 control panel will be displayed
- Click the **Configure** button on the control panel. This will load the Firefox browser and open the B092-016 Management Console



Authentication Required

Enter username and password for "127.0.0.1:8181" at http://127.0.0.1:8181

User Name:

Password:

☐ Use Password Manager to remember this password.

Cancel OK

Waiting for 127.0.0.1...

- At the Management Console menu select System: Administration

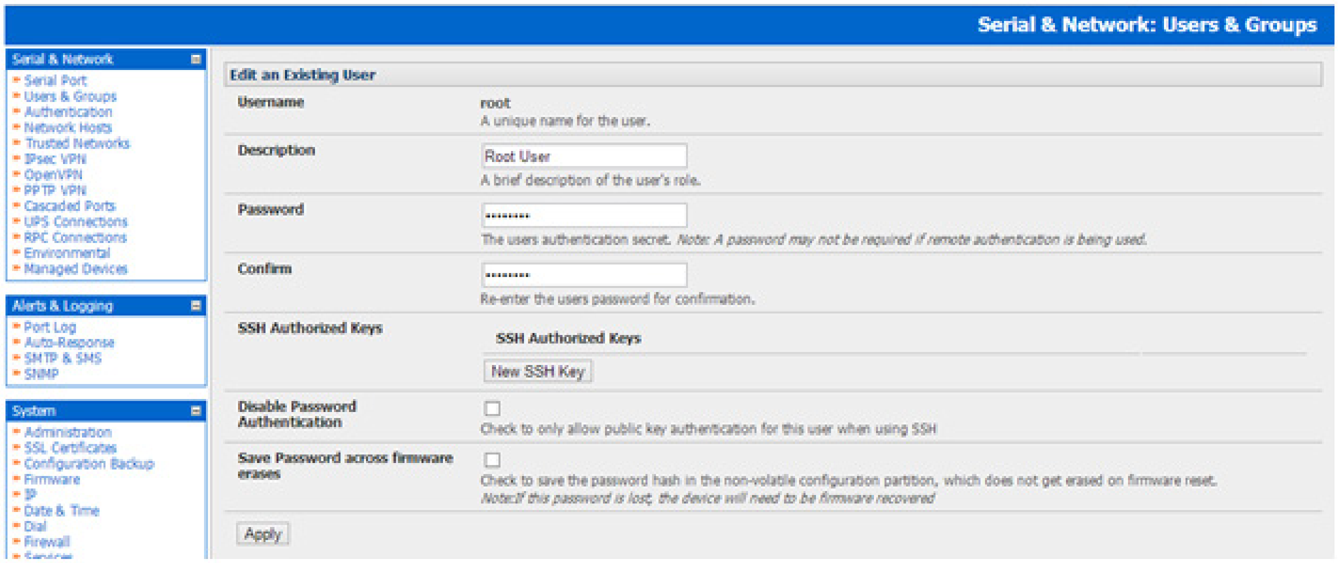
3.2 Administrator Password

For security reasons, only the administration user named **root** can initially log into your Console Server. Only those people who know the root password can access and reconfigure the Console Server itself.
However, anyone who correctly guesses the root password (and the default root password which is **default**) could gain access. It is therefore essential that you enter and confirm a new root password before giving the Console Server any access to, or control of, your computers and network appliances.

- Select ***Change default administration password*** from the **Welcome** page, which will take you to **Serial & Network: Users & Groups**
- Select **Edit** for the user *root*
- Add a new **Password** and then re-enter it in **Confirm**. This is the new password for **root**, the main administrative user account, so it is important that you choose a complex password, and keep it safe

Note*: There are no restrictions on the characters that can be used in the System Password (which can contain up to 254 characters). However, only the first eight Password characters are used to make the password hash.*

- Click **Apply**

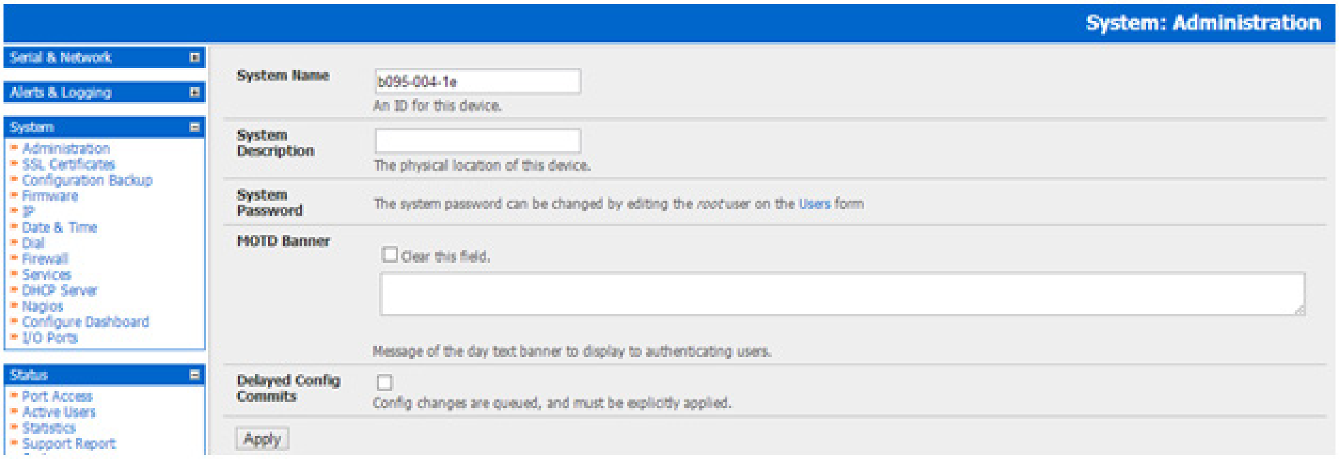


Edit an Existing User

Username	root A unique name for the user.
Description	Root User A brief description of the user's role.
Password	········ The users authentication secret. *Note: A password may not be required if remote authentication is being used.*
Confirm	········ Re-enter the users password for confirmation.
SSH Authorized Keys	SSH Authorized Keys
Disable Password Authentication	□ Check to only allow public key authentication for this user when using SSH
Save Password across firmware erases	□ Check to save the password hash in the non-volatile configuration partition, which does not get erased on firmware reset. *Note:If this password is lost, the device will need to be firmware recovered*

Note: *If the Console Server has flash memory 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. However take care as if this password is lost, the device will need to be firmware recovered.*

- Select System: Administration



System: Administration

System Name	b095-004-1e An ID for this device.
System Description	The physical location of this device.
System Password	The system password can be changed by editing the *root* user on the Users form
MOTD Banner	□ Clear this field. Message of the day text banner to display to authenticating users.
Delayed Config Commits	□ Config changes are queued, and must be explicitly applied.

- You may now wish to enter a System Name and System Description for the Console Server to give it a unique ID and make it simple to identify

Note: *The System Name can contain from 1 to 64 alphanumeric characters (however you can also use the special characters "-" "_" and "."). There are no restrictions on the characters that can be used in the System Description (which can contain up to 254 characters).*

- The MOTD Banner can be used to display a "message of the day" text to authenticating users when the ssh, ftp or web access the Console Server

- Click **Apply**. As you have changed the password you will be prompted to log in again. This time use the new password

Note*: If you are not confident your Console Server has been supplied with the current release of firmware, you can upgrade. Refer to* ***Upgrade Firmware - Chapter 10***

3.2.1 Set up new administrator

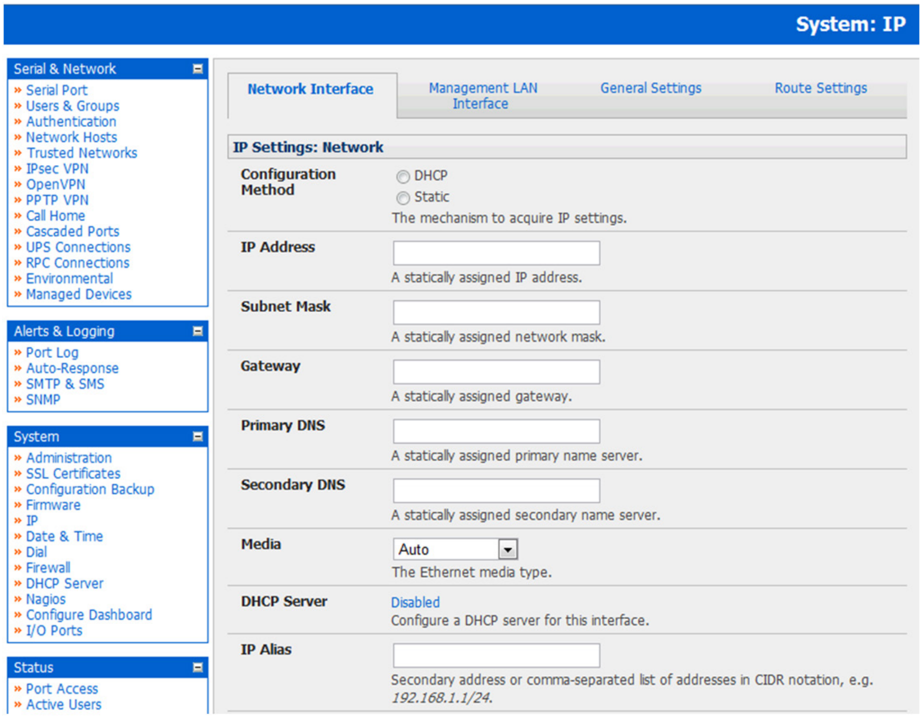
It is also recommended that you set up a new Administrator user as soon as convenient and log-in as this new user for all ongoing administration functions (rather than *root*).

This Administrator can be configured in the *admin* group with full access privileges through the **Serial & Network: Users & Groups** menu (refer to *Chapter 4* for details)

3.3 Network IP Address

It is time to enter an IP address for the principal *10/100* LAN 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 to which it is to be connected.

- On the **System: IP** menu select the **Network Interface** page then check **DHCP** or **Static** for the **Configuration Method**
- If you select **Static** you must manually enter the new **IP Address**, **Subnet Mask**, **Gateway** and **DNS** server details. This selection automatically disables the DHCP client



IP Settings: Network

Configuration Method ○ DHCP ○ Static
The mechanism to acquire IP settings.

IP Address
A statically assigned IP address.

Subnet Mask
A statically assigned network mask.

Gateway
A statically assigned gateway.

Primary DNS
A statically assigned primary name server.

Secondary DNS
A statically assigned secondary name server.

Media Auto
The Ethernet media type.

DHCP Server Disabled
Configure a DHCP server for this interface.

IP Alias
Secondary address or comma-separated list of addresses in CIDR notation, e.g. *192.168.1.1/24*.

- If you selected **DHCP** the Console Server will look for configuration details from a DHCP server on your management LAN. This selection automatically disables any static address. The Console Server MAC address can be found on a label on the base plate

Note: *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*

- By default the Console Server LAN port auto detects the Ethernet connection speed. However you can use the **Media** menu to lock the Ethernet to 10 Mb/s or 100Mb/s and to Full Duplex (FD) or Half Duplex (HD)

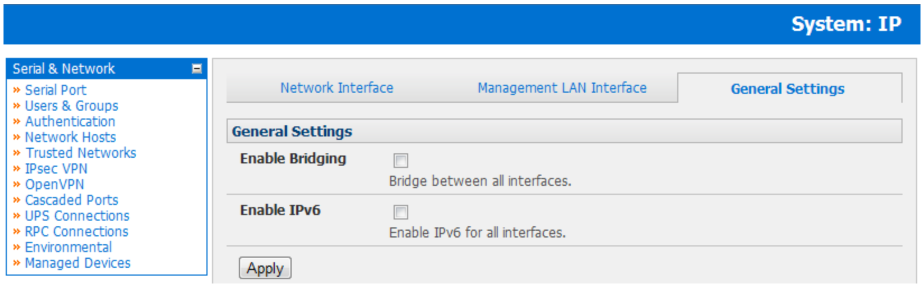
Note: *If you have changed the Console Server IP address, you may need to reconfigure your PC/workstation so it has an IP address that is in the same network range as this new address (as detailed in an earlier note in this chapter)*

- Click **Apply**
- You will need to reconnect the browser on the PC/workstation that is connected to the Console Server by entering **http://new IP address**

3.3.1 IPv6 configuration

By default, the Console Server Ethernet interfaces support IPv. However, they can also be configured for IPv6 operation:

- On the **System: IP** menu select **General Settings** page and check **Enable IPv6**



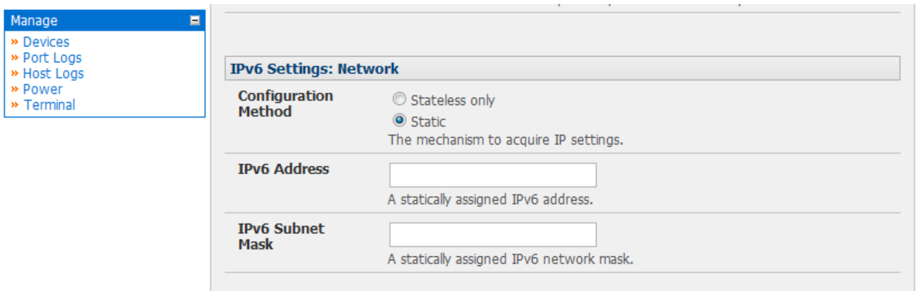
General Settings

Enable Bridging ☐
Bridge between all interfaces.

Enable IPv6 ☐
Enable IPv6 for all interfaces.

Apply

- You will then need to configure the IPv6 parameters on each network interface page



IPv6 Settings: Network

Configuration Method ○ Stateless only
● Static
The mechanism to acquire IP settings.

IPv6 Address
A statically assigned IPv6 address.

IPv6 Subnet Mask
A statically assigned IPv6 network mask.

3.3.2 Dynamic DNS (DDNS) configuration

Dynamic DNS (DDNS) enables a Console Server with a dynamically assigned IP address (that may change from time to time) to be located using a fixed host or domain name.

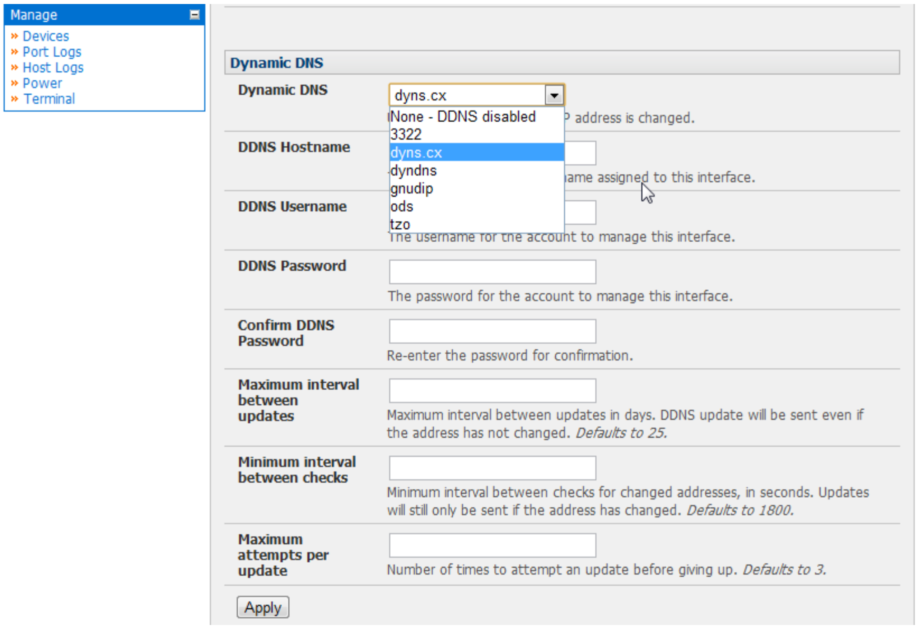
- The first step in enabling DDNS is to create an account with the supported DDNS service provider of your choice. Supported DDNS providers include:
  - o *DyNS www.dyns.cx*
  - o *dyndns.org www.dyndns.org*
  - o *GNUDip gnudip.cheapnet.net*
  - o *ODS www.ods.org*
  - o *TZO www.tzo.com*
  - o *3322.org (Chinese provider) www.3322.org*

  Upon registering with the DDNS service provider, you will 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).

  The 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



Dynamic DNS

Dynamic DNS dyns.cx
P address is changed.

DDNS Hostname
ame assigned to this interface.

DDNS Username
The username for the account to manage this interface.

DDNS Password
The password for the account to manage this interface.

Confirm DDNS Password
Re-enter the password for confirmation.

Maximum interval between updates
Maximum interval between updates in days. DDNS update will be sent even if the address has not changed. *Defaults to 25.*

Minimum interval between checks
Minimum interval between checks for changed addresses, in seconds. Updates will still only be sent if the address has changed. *Defaults to 1800.*

Maximum attempts per update
Number of times to attempt an update before giving up. *Defaults to 3.*

- In **DDNS Hostname** enter the fully qualified DNS hostname for your console server e.g. *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** i.e. the number of times to attempt an update before giving up (defaults to 3)

3.4 System Services and Service Access

Service Access specifies which access protocols/services can be used to access the Console Server (and connected serial ports).

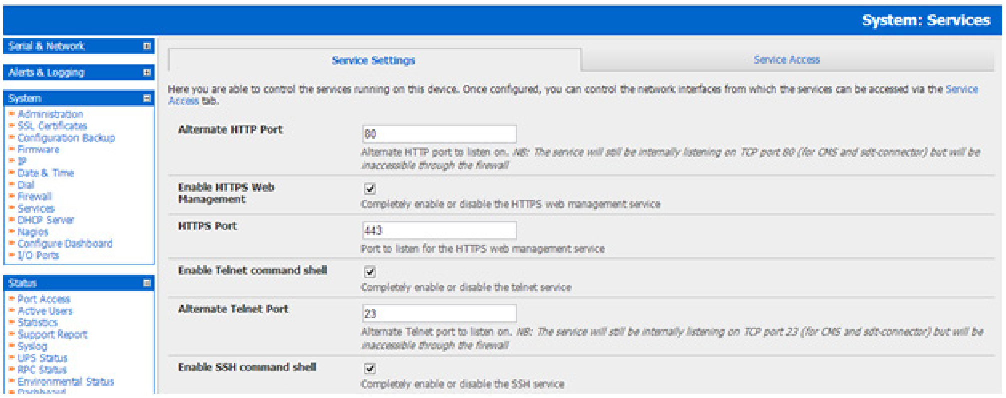
The Administrator can access and configure the Console Server (and connected devices) using a range of access protocols/services – and for each such access, the particular service must be running with access through the firewall enabled.

By default HTTP, HTTPS, Telnet and SSH services are running, and these services are enabled on all network interfaces. However, again by default, only HTTPS and SSH access to the Console Server is enabled, while HTTP and Telnet access is disabled.

For other services, such as SNMP/Nagios NRPE/NUT, the service must first be started on the relevant network interface using Service Settings. Then the Service Access can be set to allow or block access.

To enable and configure a service:

- Select the **Service Settings** tab on the **System: Services** page and enable required services



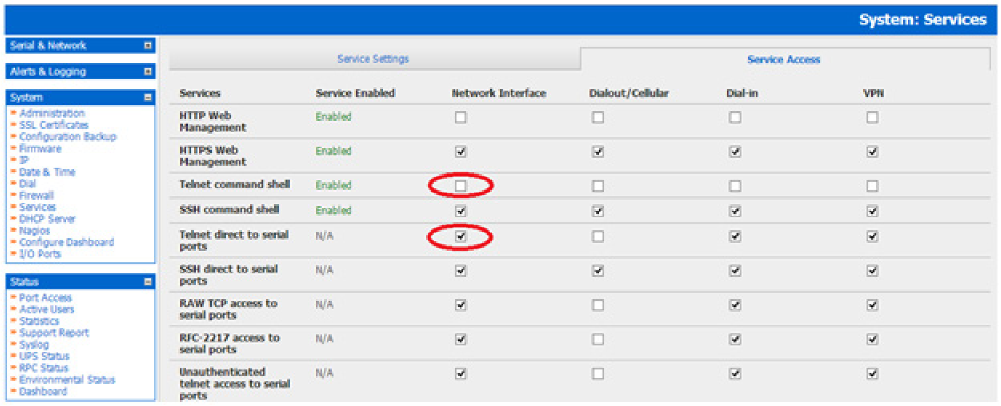
Here you are able to control the services running on this device. Once configured, you can control the network interfaces from which the services can be accessed via the Service Access tab.

Alternate HTTP Port	80 Alternate HTTP port to listen on. *NB: The service will still be internally listening on TCP port 80 (for CMS and sdt-connector) but will be inaccessible through the firewall*
Enable HTTPS Web Management	☑ Completely enable or disable the HTTPS web management service
HTTPS Port	443 Port to listen for the HTTPS web management service
Enable Telnet command shell	☑ Completely enable or disable the telnet service
Alternate Telnet Port	23 Alternate Telnet port to listen on. *NB: The service will still be internally listening on TCP port 23 (for CMS and sdt-connector) but will be inaccessible through the firewall*
Enable SSH command shell	☑ Completely enable or disable the SSH service

To change the access settings:

- Select the **Service Access** tab on the **System: Services** page. This will display the service currently enabled for the Console Server's network interfaces.
    - o Network interface (for the principal Ethernet connection)
    - o Dial out (V90 and cellular modem)
    - o Dial in (internal or external V90 modem)
    - o WiFi (802.11 wireless)
    - o OoB Failover (second Ethernet connections)
    - o VPN (IPSec or Open VPN connection over any network interface)
- Check/uncheck for each network which service access is to be enabled /disabled

In the example shown below local Administrators on local Network Interface LAN do not have Telnet access to the Console Server itself (only SSH and HTTPS access) but they do have Telnet access to the serial console devices attached to the Console Server.



Services	Service Enabled	Network Interface	Dialout/Cellular	Dial-in	VPN
HTTP Web Management	Enabled	□	□	□	□
HTTPS Web Management	Enabled	☑	☑	☑	☑
Telnet command shell	Enabled	□	□	□	□
SSH command shell	Enabled	☑	☑	☑	☑
Telnet direct to serial ports	N/A	☑	□	☑	☑
SSH direct to serial ports	N/A	☑	☑	☑	☑
RAW TCP access to serial ports	N/A	☑	□	☑	☑
RFC-2217 access to serial ports	N/A	☑	□	☑	☑
Unauthenticated telnet access to serial ports	N/A	☑	□	☑	☑

The Services Access settings specify which services the Administrator can use over which network interface to access the console server. It also nominates the enabled services that the Administrator and the User can use to connect through the Console Server to attached serial and network connected devices.

- The following general service access options can be specified:

HTTPS	This ensures the *Administrator* has secure browser access to all the Management Console 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 refer *Chapter 9 - Authentication*. By default HTTPS is enabled, and it is recommended that only HTTPS access be used if the Console Server is to be managed over any public network (e.g. the Internet).
HTTP	The HTTP service allows the Administrator basic browser access to the Management Console. It is recommended the HTTP service be disabled if the Console Server is to be remotely accessed over the Internet.
Telnet	This gives the Administrator telnet access to the system command line shell (Linux commands). While this may be suitable for a local direct connection over a management LAN, it is recommended 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
SSH	This service provides secure SSH access. It is recommended 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 PC/workstation and the SSH sever in the Console Server. For more information on SSH configuration refer *Chapter 9 - Authentication*.

- There are also a number of related service options that can be configured at this stage:

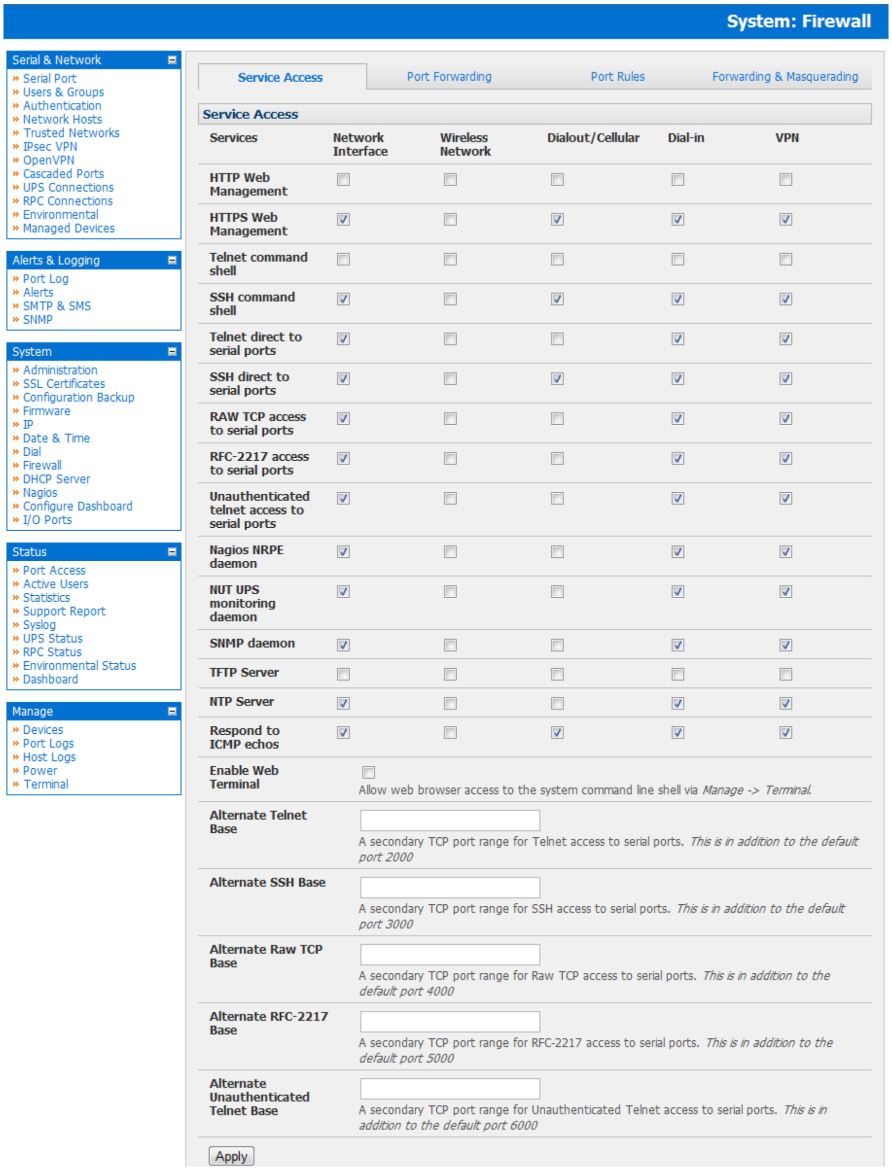
SNMP	This will enable *netsnmp* in the Console Server, which will keep a remote log of all posted information. SNMP is disabled by default. To modify the default SNMP settings, the Administrator must make the edits at the command line as described in *Chapter 15 – Advanced Configuration*
TFTP/ FTP	If a USB flash card or internal flash is detected on the Console Server, then enabling this service will set up default *tftp and ftp* servers on the USB flash. These server are used to store config files, maintain access and transaction logs etc. Files transferred using tftp will be stored under */var/tmp/usbdisk/tftpboot*
Ping	This allows the Console Server to respond to incoming ICMP echo requests. Ping is enabled by default, however for security reasons this service should generally be disabled post initial configuration
Nagios	Access to the NUT UPS monitoring and Nagios NRPE monitoring daemons
NUT	Access to the NUT UPS monitoring and Nagios NRPE monitoring daemons

- And there are some serial port access parameters that can be configured on this menu:

Base	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 (as covered in *Chapter 4 – Configuring Serial Ports*). 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 #) i.e. 2001 – 2048. So if the Administrator were to set 8000 as a secondary base for telnet then serial port #2 on the Console Server can be telnet accessed at IP Address:2002 and at IP Address:8002. The default base for SSH is 3000; for Raw TCP is 4000; and for RFC2217 it is 5000
RAW/ Direct	You can also specify that serial port devices can be accessed from nominated network interfaces using Raw TCP, direct Telnet/SSH, unauthenticated Telnet services etc

- Click Apply. As you apply your services selections, the screen will be updated with a confirmation message:

  Message Changes to configuration succeeded



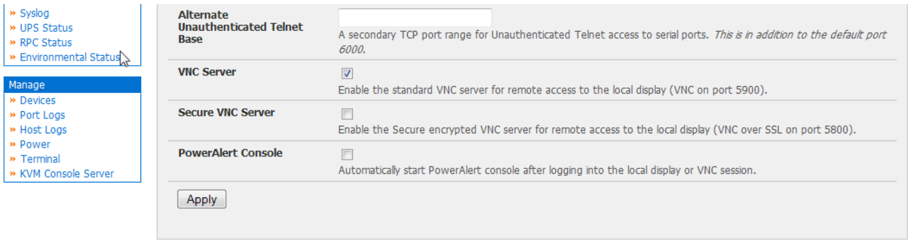
Service Access

Services	Network Interface	Wireless Network	Dialout/Cellular	Dial-in	VPN
HTTP Web Management	□	□	□	□	□
HTTPS Web Management	☑	□	☑	☑	☑
Telnet command shell	□	□	□	□	□
SSH command shell	☑	□	☑	☑	☑
Telnet direct to serial ports	☑	□	□	☑	☑
SSH direct to serial ports	☑	□	☑	☑	☑
RAW TCP access to serial ports	☑	□	□	☑	☑
RFC-2217 access to serial ports	☑	□	□	☑	☑
Unauthenticated telnet access to serial ports	☑	□	□	☑	☑
Nagios NRPE daemon	☑	□	□	☑	☑
NUT UPS monitoring daemon	☑	□	□	☑	☑
SNMP daemon	☑	□	□	☑	☑
TFTP Server	□	□	□	□	□
NTP Server	☑	□	□	☑	☑
Respond to ICMP echos	☑	□	☑	☑	☑

Enable Web Terminal	□ Allow web browser access to the system command line shell via *Manage -> Terminal.*
Alternate Telnet Base	A secondary TCP port range for Telnet access to serial ports. *This is in addition to the default port 2000*
Alternate SSH Base	A secondary TCP port range for SSH access to serial ports. *This is in addition to the default port 3000*
Alternate Raw TCP Base	A secondary TCP port range for Raw TCP access to serial ports. *This is in addition to the default port 4000*
Alternate RFC-2217 Base	A secondary TCP port range for RFC-2217 access to serial ports. *This is in addition to the default port 5000*
Alternate Unauthenticated Telnet Base	A secondary TCP port range for Unauthenticated Telnet access to serial ports. *This is in addition to the default port 6000*

- The B092-016 Console Server with PowerAlert also presents some additional service and configuration options:

VNC	The B092-016 Console Server has an internal VNC server. When enabled, it allows remote users to connect to the Console Server and run the PowerAlert software and any other embedded thin client programs as if they were plugged in locally to the KVM connectors on the B092-016 (refer to *Chapter 16* for more details). Users connect using port 5900 and need to run a VNC client applet
Secure VNC	This enables a secure encrypted remote connection using VNC over SSL on port 5800 to the B092-016 Console Server (refer to *Chapter 16*)
PowerAlert	This configuration option will automatically start the PowerAlert application on the B092-016 and display the console as soon as you log into the local display or VNC session (refer to *Chapter 16*). The complete PowerAlert manual can be downloaded at www.tripplite.com/EN/support/PowerAlert/Downloads.cfm



Alternate Unauthenticated Telnet Base	A secondary TCP port range for Unauthenticated Telnet access to serial ports. *This is in addition to the default port 6000.*
VNC Server	☑ Enable the standard VNC server for remote access to the local display (VNC on port 5900).
Secure VNC Server	□ Enable the Secure encrypted VNC server for remote access to the local display (VNC over SSL on port 5800).
PowerAlert Console	□ Automatically start PowerAlert console after logging into the local display or VNC session.

3.5 Communications Software

You need to configure the access protocols that the communications software on the Administrator and User Computer will use when connecting to the Console Server (and when connecting to serial devices and network hosts which are attached to the Console Server).

This section provides an overview of the communications software tools that can be used on the remote computer. Tripp Lite recommends the *SDT Connector* software tool that is provided with the Console Server; however, generic tools such as PuTTY and SSHTerm may also be used.

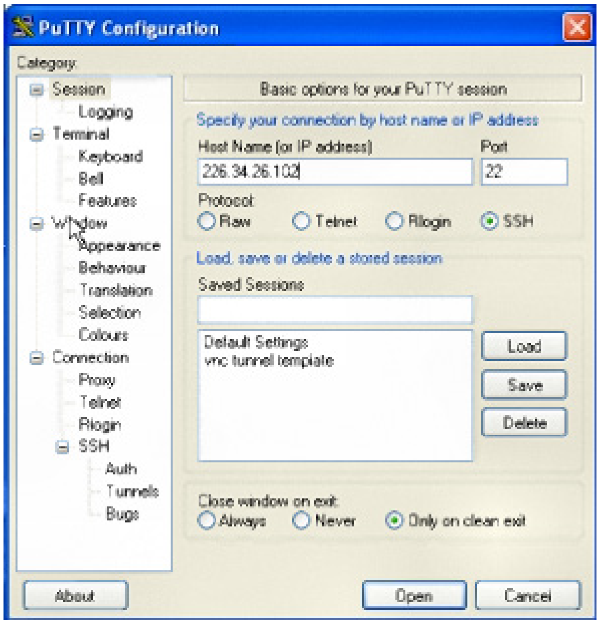
3.5.1 SDT Connector

We recommend using the *SDT Connector* communications software for all communications with Console Servers. Each Console Server is supplied with an unlimited number of *SDT Connector* licenses to use with that Console Server.

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. *SDT Connector* can be installed on Windows 2000, XP, 2003, Vista and on most Linux, UNIX and Solaris computers as detailed in *Chapter 6*.

3.5.2 PuTTY

Communications packages like *PuTTY* can be also used to connect to the Console Server command line (and to connect to serially attached devices as covered in *Chapter 4*). *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 at http://www.tucows.com/preview/195286.html



Host Name (or IP address): 226.34.26.102
Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

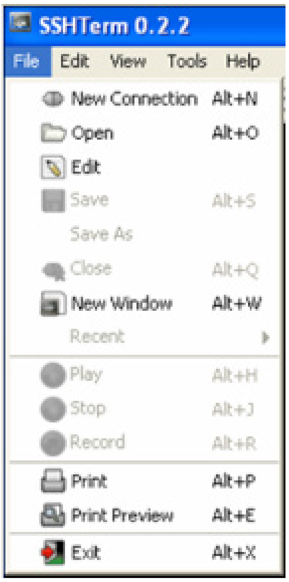
Saved Sessions:
Default Settings
vnc tunnel template

Close window on exit: ○ Always ○ Never ● Only on clean exit

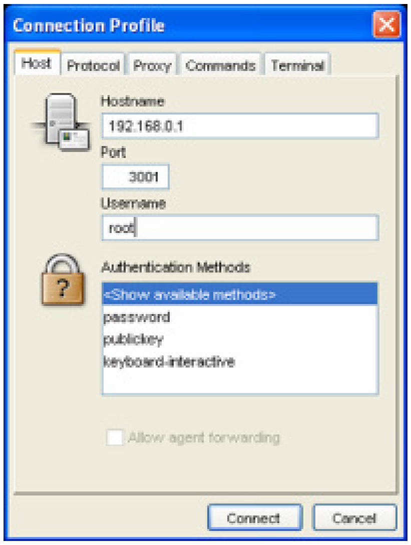
- To use PuTTY for an SSH terminal session from a Windows client, enter the Console Server's IP address as the 'Host Name (or IP address)'
- To access the Console Server command line, select 'SSH' as the protocol and use the default IP Port 22
- Click 'Open' and the Console Server login prompt will appear. (You may also receive a 'Security Alert' that the host's key is not cached. Choose 'yes' to continue.)
- Using the Telnet protocol is similarly simple, but you need to use the default port 23

3.5.3 SSHTerm

Another common communications package that may be useful is *SSHTerm*. This is 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, simply Select the 'File' option and click on 'New Connection'.
- A new dialog box will appear for your 'Connection Profile'. Type in the host name or IP address (for the Console Server unit) and the TCP port that the SSH session will use (port 22). Then type in your username and choose password authentication and click Connect.



Hostname
192.168.0.1
Port
3001
Username
root
Authentication Methods
<Show available methods>
password
publickey
keyboard-interactive

□ Allow agent forwarding

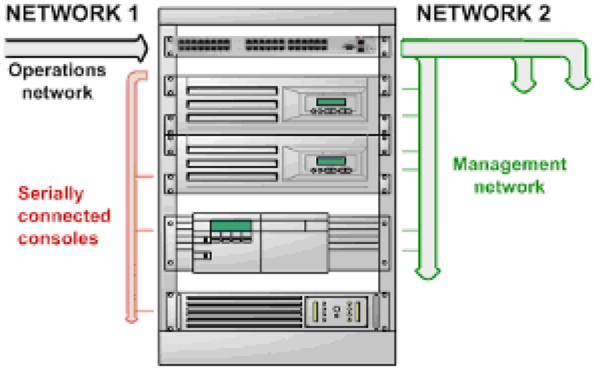
- A message may appear about the host key fingerprint. You will need to select 'Yes' or 'Always' to continue.
- The next step is password authentication. You will be prompted for your username and password from the remote system. You will then be logged on to the Console Server

3.6 Management Network Configuration

The B096-048/032/016 Console Server Management Switches and B094-008-2E-M-F / B094-008-2E-V Console Server each have an additional network port that can be configured as a Management LAN port or as a failover/ OOB access port.

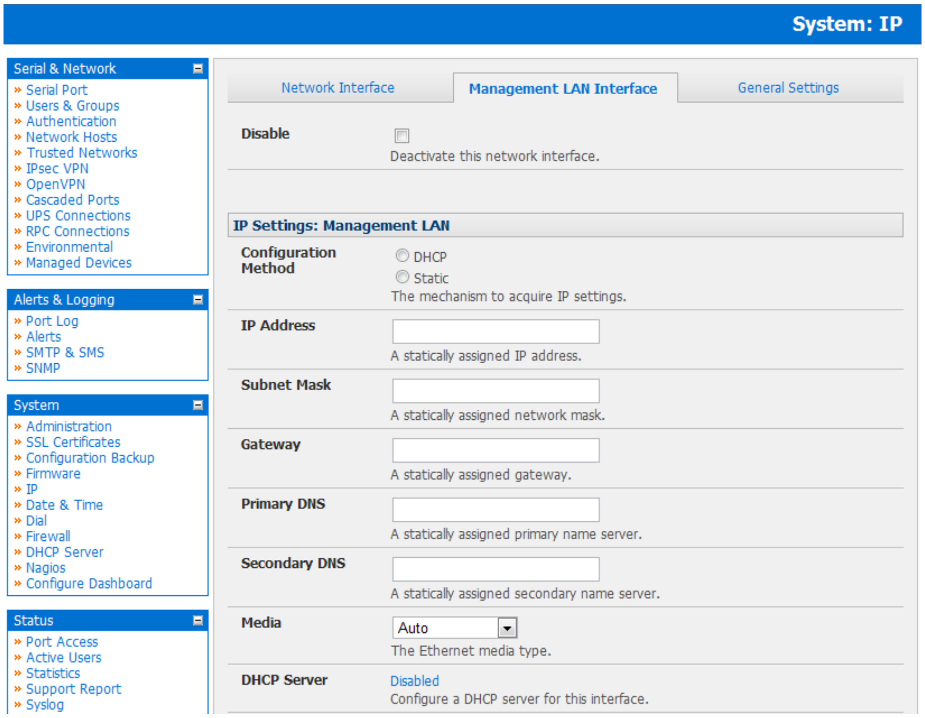
3.6.1 Enable the Management LAN

The B096-048/032/016 Console Server Management Switches and B094-008-2E-M-F / B094-008-2E-V Console Server have dual Ethernet ports which can be configured to provide a management LAN gateway. With this configuration, the B096-048/032/016 and B094-008-2E-M-F / B094-008-2E-V provide firewall, router and DHCP server features and you can connect managed hosts to this management LAN.



These features are all disabled by default. To configure the Management LAN gateway:

- Select the **Management LAN Interface** page on the **System: IP** menu and uncheck **Disable**
- Configure the **IP Address** and **Subnet Mask** for the Management LAN (but leave the **DNS** fields blank)
- Click **Apply**



Disable □
Deactivate this network interface.

IP Settings: Management LAN

Configuration Method ○ DHCP ○ Static
The mechanism to acquire IP settings.

IP Address
A statically assigned IP address.

Subnet Mask
A statically assigned network mask.

Gateway
A statically assigned gateway.

Primary DNS
A statically assigned primary name server.

Secondary DNS
A statically assigned secondary name server.

Media Auto
The Ethernet media type.

DHCP Server Disabled
Configure a DHCP server for this interface.

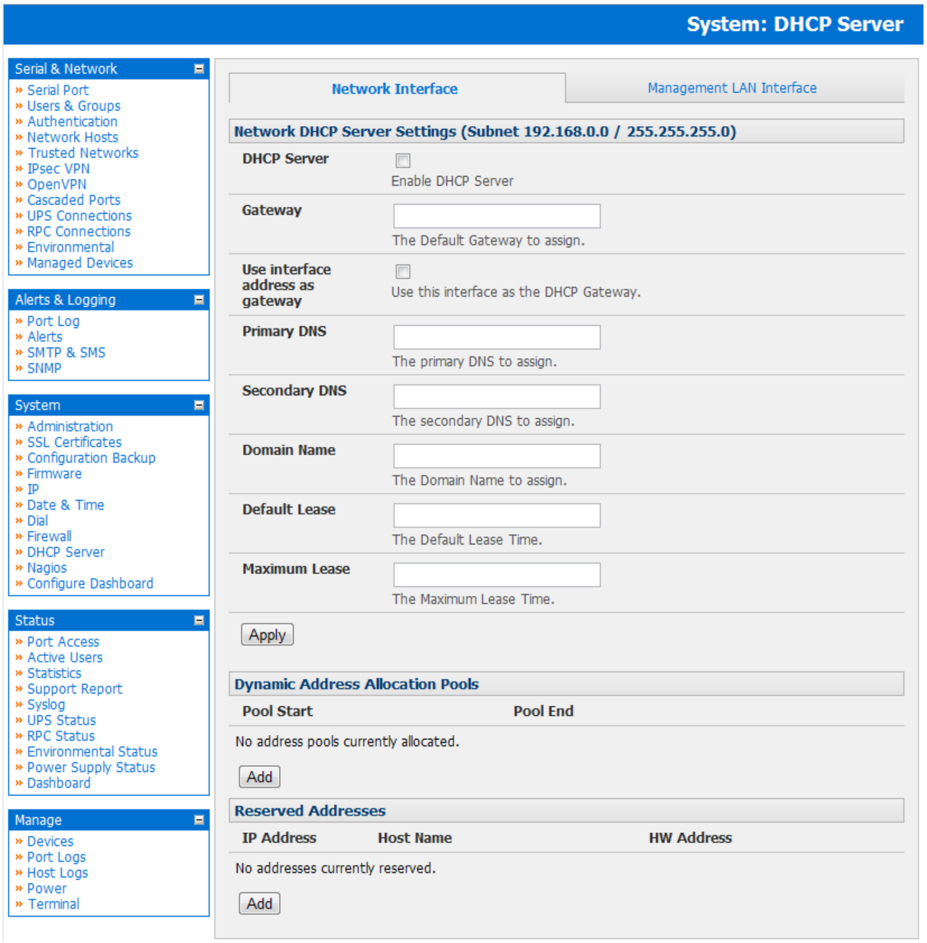
***Note**: With the B094-008-2E-M-F, B096-048, B094-008-2E-V, B096-032 and B096-016 the second Ethernet port can be configured as either a gateway port or it can be configured as an OOB/Failover port - but not both. So ensure you did not allocate the **Management LAN** as the **Failover Interface** when you configured the principal **Network** connection on the **System: IP** menu*

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.

3.6.2 Configure the DHCP server

The Console Servers also host a DHCP server which by default is disabled. The DHCP server enables the automatic distribution of IP addresses to devices on the Network Interface or the Management LAN. To enable the DHCP server:

- On the **System: IP** menu select the **Management LAN Interface** page and click the **Disabled** label in the **DHCP Server** field (or go to the **System: DHCP Server** menu and check **Enable DHCP Server**)



Network DHCP Server Settings (Subnet 192.168.0.0 / 255.255.255.0)

DHCP Server	□ Enable DHCP Server
Gateway	The Default Gateway to assign.
Use interface address as gateway	□ Use this interface as the DHCP Gateway.
Primary DNS	The primary DNS to assign.
Secondary DNS	The secondary DNS to assign.
Domain Name	The Domain Name to assign.
Default Lease	The Default Lease Time.
Maximum Lease	The Maximum Lease Time.

Dynamic Address Allocation Pools

Pool Start	Pool End

No address pools currently allocated.

Reserved Addresses

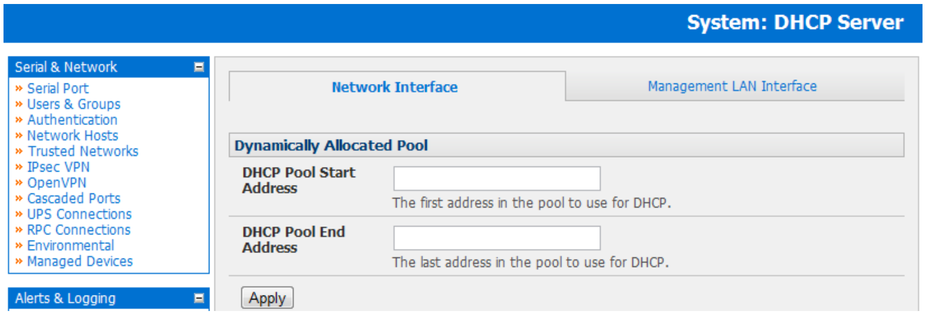
IP Address	Host Name	HW Address

No addresses currently reserved.

- Enter the **Gateway** address that is to be issued to the DHCP clients. If this field is left blank, the Console Server's IP address will be used
- Enter the **Primary DNS** and **Secondary DNS** address to issue the DHCP clients. Again if this field is left blank, Console Server's IP address is used, so leave this field blank for automatic DNS server assignment
- Optionally enter a **Domain Name** suffix to issue DHCP clients
- 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 will sequentially issue IP addresses from a specified address pool(s):

- Click **Add** in the **Dynamic Address Allocation Pools** field
- Enter the **DHCP Pool Start Address** and **End Address** and click **Apply**



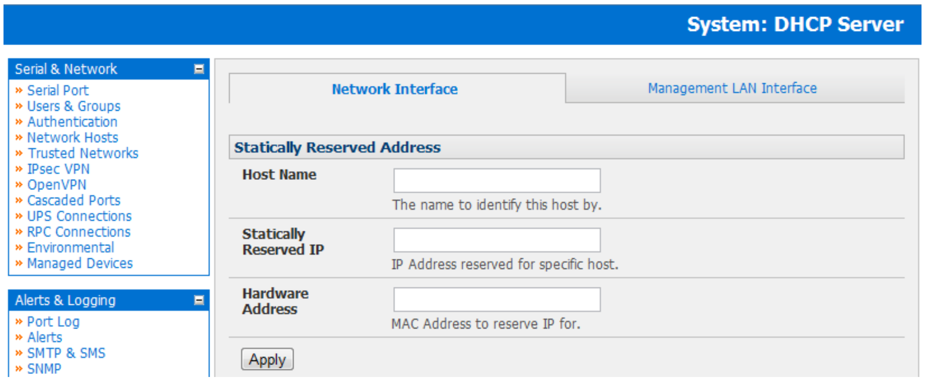
Dynamically Allocated Pool

DHCP Pool Start Address: The first address in the pool to use for DHCP.

DHCP Pool End Address: The last address in the pool to use for DHCP.

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 and click **Apply**



Statically Reserved Address

Host Name: The name to identify this host by.

Statically Reserved IP: IP Address reserved for specific host.

Hardware Address: MAC Address to reserve IP for.

When DHCP has initially allocated hosts addresses it is recommended to copy these into the pre-assigned list so the same IP address will be reallocated in the event of a reboot.

3.6.3 Select Failover or broadband OOB

The Console Servers provide a failover option so in the event of a problem using the main LAN connection for accessing the Console Server; an alternate access path is automatically used.

- By default the failover is not enabled. To enable, select the **Network Interface** page on the **System: IP** menu
- Now select the **Failover Interface** to be used in the event of an outage on the main network. This can be:
  - o a second Ethernet connection on the B094-008-2E-M-F / B094-008-2E-V or B096-048/032/016
  - o the B094-008-2E-M-F / B094-008-2E-V or B096-048/032/016 internal modem
  - o an external modem device connected to the Console Server
- Click **Apply**. You have selected the failover method. However it is not active until you have specified the external sites to be probed to trigger failover, and set up the failover ports themselves. This is covered in *Chapter 5*.

Note*: The second Ethernet port on the* B094-008-2E-M-F / B094-008-2E-V or *B096-048/032/016 can be configured as either a Management LAN gateway port or it can be configured as an OoB/Failover port - but not both. So ensure you did not configure this port as the* ***Management LAN*** *on the* ***System: IP*** *menu*

3.6.4 Bridging the network ports

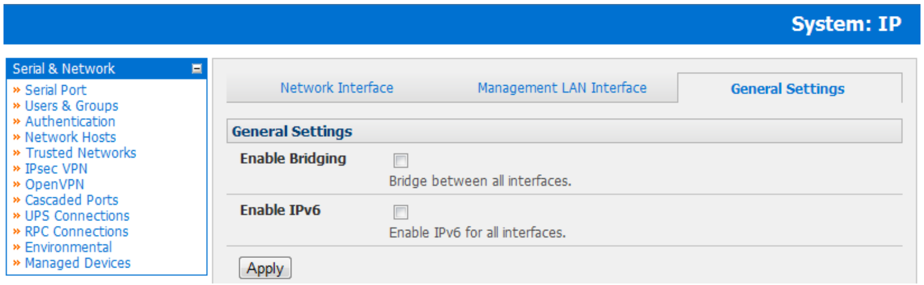
By default the B096-048/032/016 Console Server's Management LAN network port can only be accessed using SSH tunneling /port forwarding or by establishing an IPsec VPN tunnel to the Console Server.

However the network ports on the Console Servers can be bridged.

- Select **Enable Bridging** on the **System: IP General Settings** menu

With bridging enabled:

- the Ethernet ports are transparently interconnected at the data link layer (layer 2)
- the Ethernet ports are configured collectively using the **Network Interface** menu
- network traffic is forwarded between all Ethernet ports with no firewall restrictions
- the **Management LAN Interface** and **Out-of-Band/Failover Interface** functions are removed and the **DHCP Server** is disabled



General Settings

Enable Bridging □ Bridge between all interfaces.

Enable IPv6 □ Enable IPv6 for all interfaces.

An alternate to bridging is to use the firewall/routing functions (packet filtering, port forwarding, masquerading) functions detailed in chapter 5. This can provide firewalled remote IP access to devices on the Management LAN.

3.6.5 Wireless LAN

Console Servers can be fitted externally with an external 802.11 wireless USB dongle. The wireless device will then be auto-detected on power up and you will be presented with a **Wireless LAN Interface** menu in the **System: IP** menu

- The wireless LAN is deactivated by default so to activate it first uncheck Disable

To configure the IP settings of the wireless LAN:

- Select DHCP or Static for the Configuration Method
  - o If you selected **Static** then manually enter the new **IP Address**, **Subnet Mask**, **Gateway** and **DNS** server details. This selection automatically disables the DHCP client
  - o If you selected **DHCP** the Console Server will look for configuration details from a DHCP server on your management LAN. This selection automatically disables any static address. The Console Server MAC address can be found on a label on the base plate
- The wireless LAN when enabled will operate as the main network connection to the *console server* so failover is available (though it not *enabled* by default). Use **Failover Interface** to select the device to failover to in case of wireless outage and specify **Probe Addresses** of the peers to probed for connectivity detection
- Configure the Wireless Client to select the local wireless network which will serve as the main network connection to the Console Server.
  - o Enter the appropriate **SSID** (Set Service Identifier) of the wireless access point to connect to
  - o Select the **Wireless Network Type** where **Infrastructure** is used to connect to an access point and **Ad-hoc** to connect directly to a computer
  - o Select the **Wireless Security** mode of the wireless network (WEP, WPA etc) and enter the required Key/ Authentication/ Encryption settings

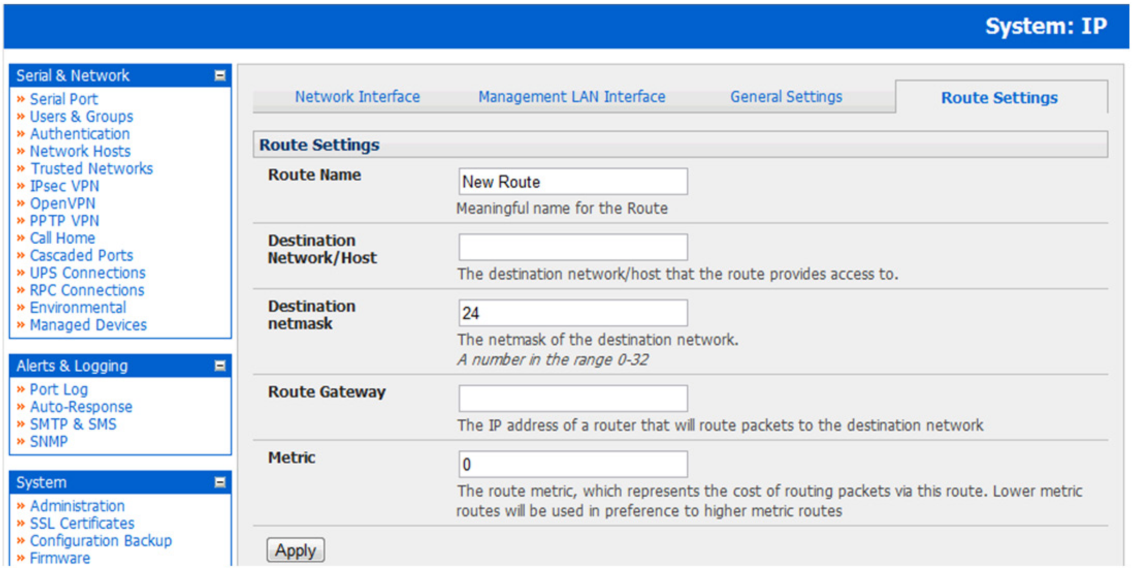
Note: *The* Wireless *screen in* Status: Statistics *will display all the locally accessible wireless LANs (with SSID and Encryption/ Authentication settings). You can also use this screen to confirm you have successfully connected to the selected access point.*

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. This chapter covers each of the steps in configuring hosts and serially attached devices:

- *Configure Serial Ports* – setting up the protocols to be used in accessing serially-connected devices
- *Users & Groups* – setting up users and defining the access permissions for each of these users
- *Authentication* – this is covered in more detail in Chapter 9
- *Network Hosts* – configuring access to local network connected computers or appliances (hosts)
- *Configuring Trusted Networks* - nominate specific IP addresses that trusted users access from
- *Cascading and Redirection of Serial Console Ports*
- *Connecting to Power (UPS PDU and IPMI) and Environmental Monitoring (EMD) devices*
- *Serial Port Redirection* – using the VirtualPort windows and Linux clients
- *Managed Devices* - presents a consolidated view of all the connections
- *IPSec* – enabling IPSec VPN connection
- *OpenVPN* - enabling IPSec OpenVPN connection
- *PPTP* – setting up point to point connection

3.6.6 Static routes

Firmware 3.4 and later support static routes which provide a very quick way to route data from one subnet to another. You can hard code a path that specifies to the console server/router which path to take to get to a particular subnet. This may be useful for remotely accessing various subnets at a remote site when using the cellular OoB connection.



Route Settings

Route Name	New Route Meaningful name for the Route
Destination Network/Host	 The destination network/host that the route provides access to.
Destination netmask	24 The netmask of the destination network. *A number in the range 0-32*
Route Gateway	 The IP address of a router that will route packets to the destination network
Metric	0 The route metric, which represents the cost of routing packets via this route. Lower metric routes will be used in preference to higher metric routes

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

- Select the **Route Settings** tab on the **System: IP General Settings** menu
- Enter a meaningful **Route Name** for the route
- In the **Destination Network/Host** field, enter the IP address of the destination network/host that the route provides access to
- Enter a value in the **Destination netmask** field that identifies the destination network or host. Use any number between 0 and 32. A subnet mask of 32 identifies a host route.
- In the **Route Gateway** field, enter the IP address of a router that will route packets to the destination network
- 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. Use any number equal to or greater than 0
- Click **Apply**

4.1 Configuring Serial Ports

To configure a serial port you must first set the **Common Settings** (*Chapter 4.1.1*) that are to be used for the data connection to that port (e.g. baud rate) and the mode the port is to operate in. Each port can be set to support one of five operating modes:

i. **Console Server Mode** (*Chapter 4.1.2*) is the default setting and enables general access to the serial console port on serially attached devices

ii. **Device Mode** (*Chapter 4.1.3*) sets the serial port up to communicate with an intelligent serial controlled PDU, UPS or Environmental Monitor Devices (EMD)

iii. **SDT Mode** (*Chapter 4.1.4*) enables graphical console access (with RDP, VNC, HTTPS etc) to hosts that are serially connected

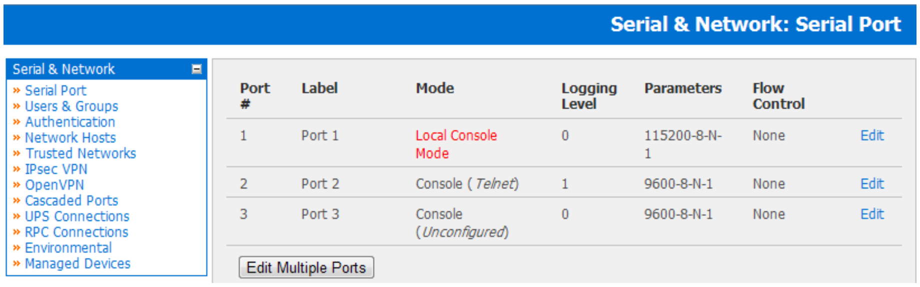
iv. **Terminal Server Mode** (*Chapter 4.1.5*) sets the serial port to await an incoming terminal login session

v. **Serial Bridge Mode** (*Chapter 4.1.6*) enables the transparent interconnection of two serial port devices over a network

To select the serial port to configure:

- Select Serial & Network: Serial Port and click Edit on the port to be reconfigured

***Note**: If you wish 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*



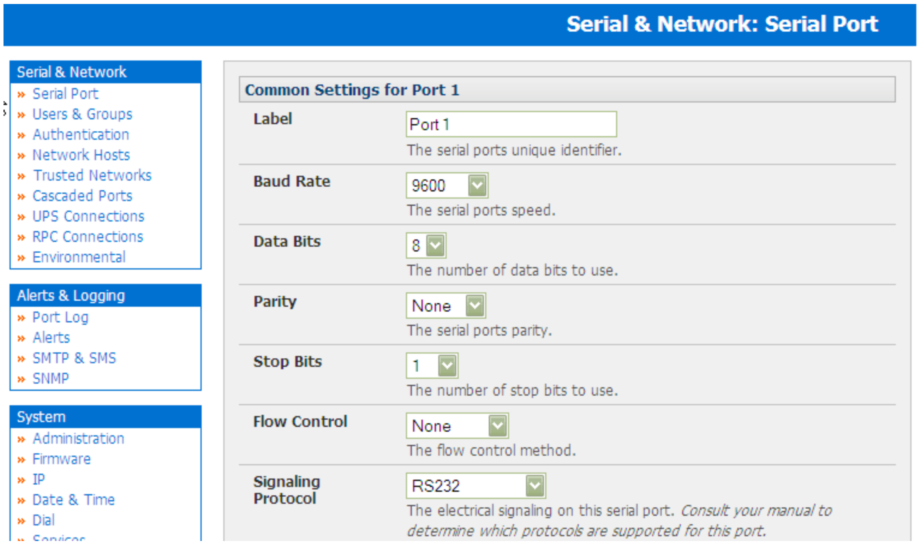
Port #	Label	Mode	Logging Level	Parameters	Flow Control	
1	Port 1	Local Console Mode	0	115200-8-N-1	None	Edit
2	Port 2	Console (*Telnet*)	1	9600-8-N-1	None	Edit
3	Port 3	Console (*Unconfigured*)	0	9600-8-N-1	None	Edit

- When you have configured the common settings and the mode for each port, set up any remote syslog (*Chapter 4.1.7*), then click **Apply**

- 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 (refer to *Chapter 10 – Nagios Integration*)

4.1.1 Common Settings

There are a number of common settings available 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 which is attached to that port.



Common Settings for Port 1

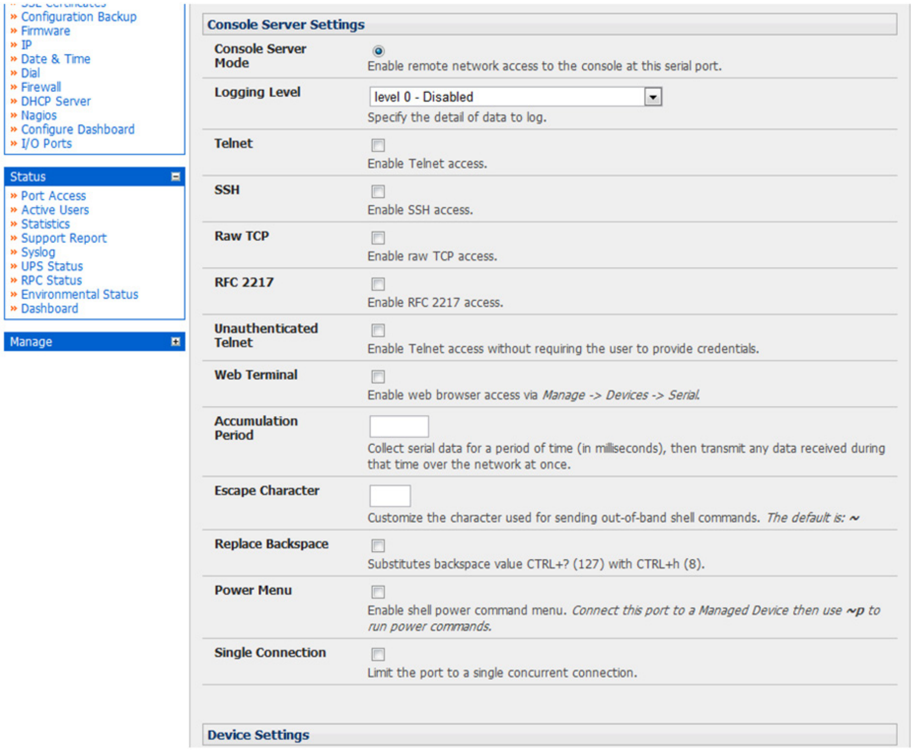
Label	Port 1 The serial ports unique identifier.
Baud Rate	9600 The serial ports speed.
Data Bits	8 The number of data bits to use.
Parity	None The serial ports parity.
Stop Bits	1 The number of stop bits to use.
Flow Control	None The flow control method.
Signaling Protocol	RS232 The electrical signaling on this serial port. *Consult your manual to determine which protocols are supported for this port.*

- Select Serial & Network: Serial Port and click Edit
- Specify a label for the port
- Select the appropriate Baud Rate, Parity, Data Bits, Stop Bits and Flow Control for each port (and ensure they match the settings for serial device that is connected). The Signaling Protocol is hard configured to be RS232

***Note**: The serial ports are all set at the factory to RS232 9600 baud, no parity, 8 data bits, 1 stop bit and Console Server Mode. The baud rate can be changed to 2400 – 230400 baud using the management console. Lower baud rates (50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800 baud) can be configured from the command line as detailed in Chapter 14*

4.1.2 Console Server Mode

Select **Console Server Mode** to enable remote management access to the serial console that is attached to the serial port:



Console Server Settings

Console Server Mode ● Enable remote network access to the console at this serial port.

Logging Level level 0 - Disabled — Specify the detail of data to log.

Telnet □ Enable Telnet access.

SSH □ Enable SSH access.

Raw TCP □ Enable raw TCP access.

RFC 2217 □ Enable RFC 2217 access.

Unauthenticated Telnet □ Enable Telnet access without requiring the user to provide credentials.

Web Terminal □ Enable web browser access via *Manage -> Devices -> Serial.*

Accumulation Period Collect serial data for a period of time (in milliseconds), then transmit any data received during that time over the network at once.

Escape Character Customize the character used for sending out-of-band shell commands. *The default is:* ~

Replace Backspace □ Substitutes backspace value CTRL+? (127) with CTRL+h (8).

Power Menu □ Enable shell power command menu. *Connect this port to a Managed Device then use* **~p** *to run power commands.*

Single Connection □ Limit the port to a single concurrent connection.

Device Settings

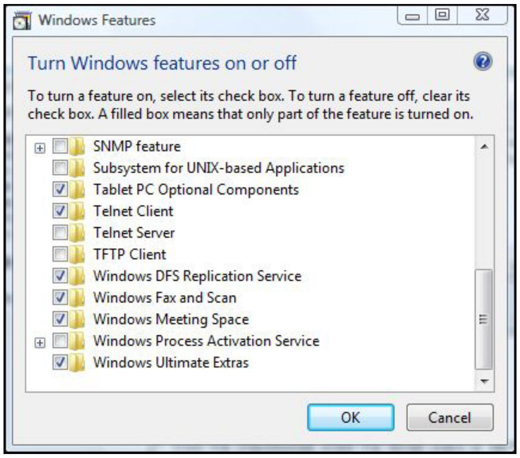
Logging Level This specifies the level of information to be logged and monitored (refer to *Chapter 7 - Alerts and Logging*)

Telnet Check to enable Telnet access to the serial port. When enabled, a Telnet client on a User or Administrator's computer can connect to a serial device attached to this serial port on the Console Server. The default port address is IP Address _ Port (2000 + serial port #) i.e. 2001 – 2048

Telnet communications are unencrypted, so this protocol is generally recommended for local connections only. However, if the remote communications are being tunneled with *SDT Connector*, then Telnet can be used to securely access these attached devices (*see Note below*).

With Win2000/XP/NT you can run Telnet from the command prompt (*cmd.exe*). Vista comes with a Telnet client and server but they are not enabled by default. To enable Telnet, simply:

- Log in as *Admin* and go to *Start/ Control Panel/Programs and Features*
- Select *Turn Windows Features On or Off*, check the *Telnet Client* and click *OK*



Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ☐ SNMP feature
- ☐ Subsystem for UNIX-based Applications
- ☑ Tablet PC Optional Components
- ☑ Telnet Client
- ☐ Telnet Server
- ☐ TFTP Client
- ☑ Windows DFS Replication Service
- ☑ Windows Fax and Scan
- ☑ Windows Meeting Space
- ☐ Windows Process Activation Service
- ☑ Windows Ultimate Extras

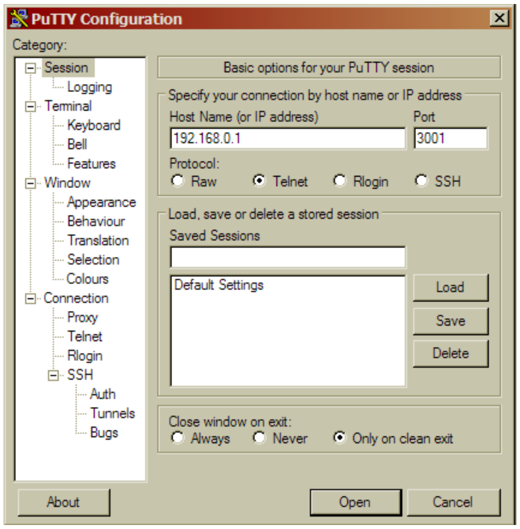
***Note**: In Console Server mode, Users and Administrators can use* SDT Connector *to set up secure Telnet connections that are SSH tunneled from their client computers to the serial port on the Console Server with a simple point-and-click.*

To use SDT Connector *to access consoles on the Console Server serial ports, configure the* SDT Connector *with the Console Server as a* gateway*, then as a* host*. Now enable Telnet service on Port (2000 + serial port #) i.e. 2001–2048. Refer to* Chapter 6 *for more details on using* SDT Connector *for Telnet and SSH access to devices attached to the Console Server serial ports.*

You can also use standard communications packages like *PuTTY* to set a direct Telnet (or SSH) connection to the serial ports (refer Note below):

***Note**: PuTTY also supports Telnet (and SSH). The procedure to set up a Telnet session is simple: 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 (i.e. 2001 to 2048).*

Click the 'Open' button. You may then receive a 'Security Alert' that the host's key is not cached. Choose 'yes' to continue. You will then be presented with the login prompt of the remote system connected to the serial port chosen on the Console Server. You can login as normal and use the host serial console screen.



PuTTY Configuration

Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address): 192.168.0.1

Port: 3001

Protocol: ○ Raw ● Telnet ○ Rlogin ○ SSH

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

PuTTY can be downloaded at http://www.tucows.com/preview/195286.html

SSH

It is recommended that the User or Administrator uses SSH as the protocol for connecting to serial consoles attached to the Console Server when communicating over the Internet or any other public network. This will provide an authenticated, encrypted connection between the SSH client program on the remote user's computer and the Console Server. The user's communication with the serial device attached to the Console Server is therefore secure.

It is recommended for Users and Administrators to use SDT Connector when making an SSH connection to the consoles on devices attached to the Console Server's serial ports. Configure the SDT Connector with the Console Server as a gateway, then as a host, and enable SSH service on Port (3000 + serial port #) i.e. 3001-3048 (refer to *Chapter 6*).

You can also use common communications packages, like *PuTTY* or *SSHTerm* to SSH connect directly to port address IP Address _ Port (3000 + serial port #) i.e. 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:

<username>:*<portXX>*

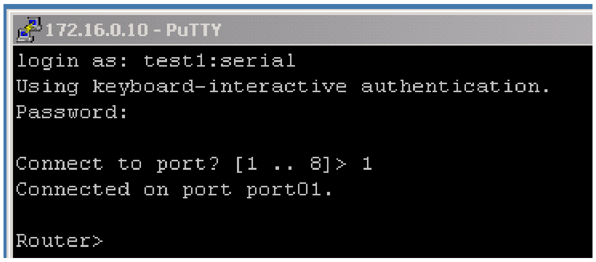
<username>:*<port label>*

<username>:*<ttySX>*

<username>:*<serial>*

So for a user named 'fred' to access serial port 2, when setting up the SSHTerm or the PuTTY SSH client, instead of typing *username = fred* and *ssh port = 3002*, the alternate is to type *username = fred:port02* (or *username = fred:ttyS1*) and *ssh port = 22*.

Or, by typing *username=fred:serial* and *ssh port = 22*, the user is presented with a port selection option:



```
login as: test1:serial
Using keyboard-interactive authentication.
Password:

Connect to port? [1 .. 8]> 1
Connected on port port01.

Router>
```

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/gateway.

TCP

RAW TCP allows connections directly to a TCP socket. Communications programs such as *PuTTY* also support RAW TCP; however, this protocol would usually be used by a custom application. For RAW TCP, the default port address is IP Address _ Port (4000 + serial port #) i.e. 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 interconnected over a network (see *Chapter 4.1.6 – Serial Bridging*).

RFC2217

Selecting *RFC2217* enables serial port redirection on that port. For RFC2217, the default port address is IP Address _ Port (5000 + serial port #) i.e. 5001 – 5048.

You will also need to run serial port redirector software on your desktop computer. This software, which supports RFC2217 virtual com ports, is available commercially and as freeware, for Windows UNIX and Linux, and it allows you to use a serial device connected to the remote Console Server as if it were connected to your local serial port.

Unauthenticated Telnet Selecting *Unauthenticated Telnet* enables Telnet access to the serial port without requiring the user to provide credentials. When a user accesses the Console Server to Telnet to a serial port they are normally given a login prompt. However, with unauthenticated Telnet, they connect directly through to port with any Console Server login at all. 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.

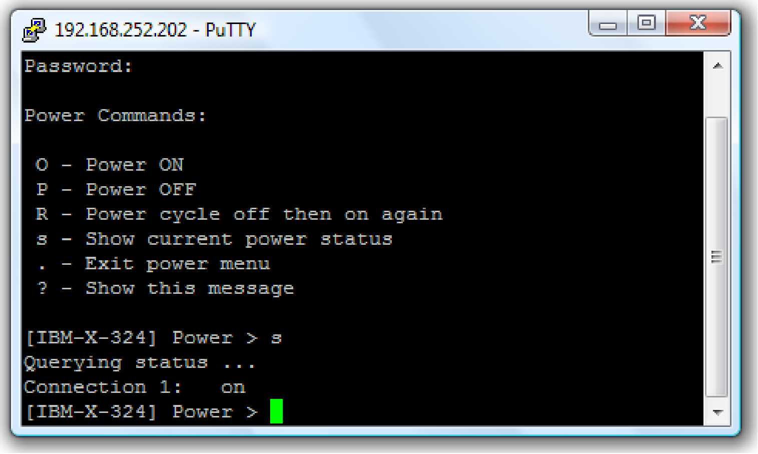
For Unauthenticated Telnet, the default port address is IP Address _ Port (6000 + serial port #) i.e. 6001 – 6048.

Web Terminal Selecting 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 13.3 for more details.

Accumulation Period By default once a connection has been established for a particular serial port (such as a RFC2217 redirection or Telnet connection to a remote computer) then 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

Escape Character This enables you to change the character used for sending escape characters. The default is ~.

Power Menu This setting enables the shell power command so a user can control the power connection to a Managed Device from command line when they are telnet or SSH connected to the device. To operate the Managed Device must be set up with both its Serial port connection and Power connection configured. The command to bring up the power menu is ~p

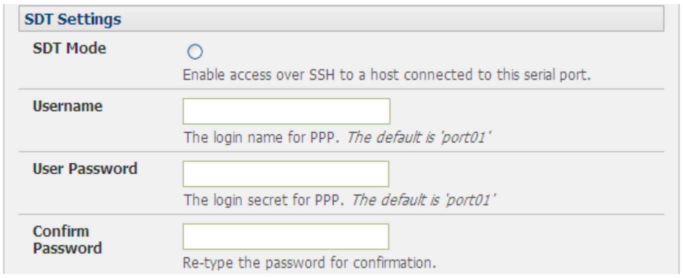


```
Password:

Power Commands:

 O - Power ON
 P - Power OFF
 R - Power cycle off then on again
 s - Show current power status
 . - Exit power menu
 ? - Show this message

[IBM-X-324] Power > s
Querying status ...
Connection 1:   on
[IBM-X-324] Power >
```

Single Connection This setting limits the port to a single connection so if multiple users have access privileges for a particular port only one user at a time can be accessing that port (i.e. port “snooping” is not permitted)

4.1.3 SDT Mode

This setting allows port forwarding of LAN protocols such as RDP, VNC, HTPP, HTTPS, SSH and Telnet through to computers which are connected locally 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.



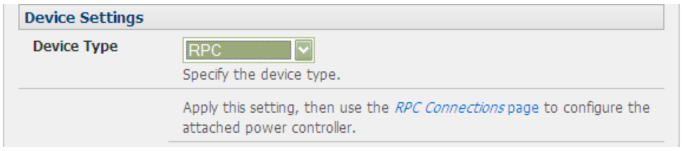
SDT Settings

SDT Mode	○ Enable access over SSH to a host connected to this serial port.
Username	 The login name for PPP. *The default is 'port01'*
User Password	 The login secret for PPP. *The default is 'port01'*
Confirm Password	 Re-type the password for confirmation.

Refer to *Chapter 6.6 - **Using SDT Connector to Telnet or SSH connect to devices that are serially attached to the Console Server*** for configuration details

4.1.4 Device (RPC, UPS, EMD) Mode

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



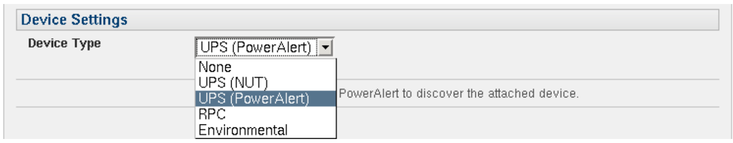
Device Settings

Device Type: RPC

Specify the device type.

Apply this setting, then use the RPC Connections page to configure the attached power controller.

- 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 8 - Power & Environmental Management.* The B092-016 Console Server also allows you to configure ports as UPS devices that PowerAlert will manage. PowerAlert will discover the attached UPS device and auto-configure. See www.tripplite.com/EN/support/PowerAlert/Downloads.cfm for a complete PowerAlert manual.



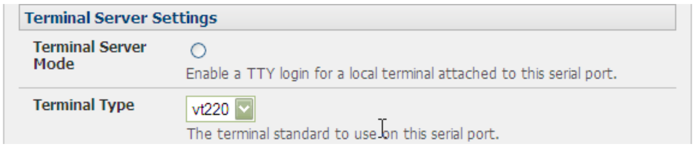
Device Settings

Device Type: UPS (PowerAlert)

PowerAlert to discover the attached device.

4.1.5 Terminal Server Mode

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



Terminal Server Settings

Terminal Server Mode: ○

Enable a TTY login for a local terminal attached to this serial port.

Terminal Type: vt220

The terminal standard to use on this serial port.

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.

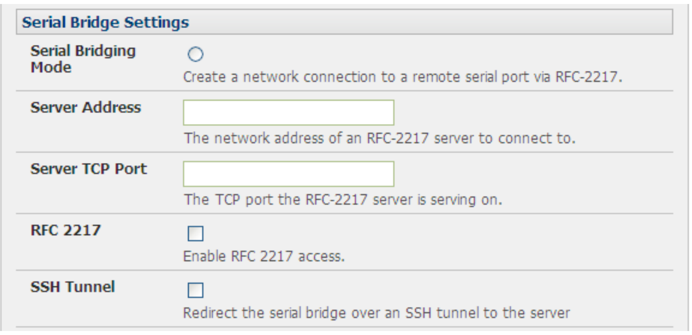
***Note**: Selecting Terminal Server mode will disable Port Manager for that serial port, so data is no longer logged for alerts etc.*

4.1.6 Serial Bridging Mode

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.

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 *Chapter 4.1.2 – Console Server Mode*).

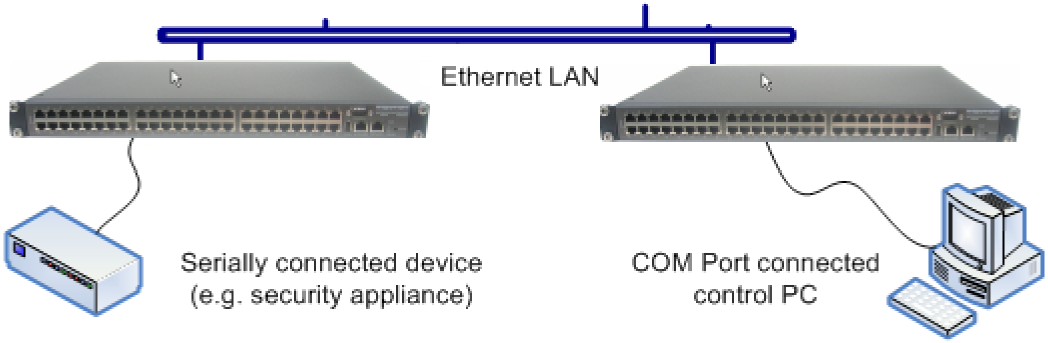
For the Client Console Server, the serial port to be bridged must be set in Bridging Mode:



Serial Bridge Settings

Serial Bridging Mode	○ Create a network connection to a remote serial port via RFC-2217.
Server Address	The network address of an RFC-2217 server to connect to.
Server TCP Port	The TCP port the RFC-2217 server is serving on.
RFC 2217	□ Enable RFC 2217 access.
SSH Tunnel	□ Redirect the serial bridge over an SSH tunnel to the server

- Select **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

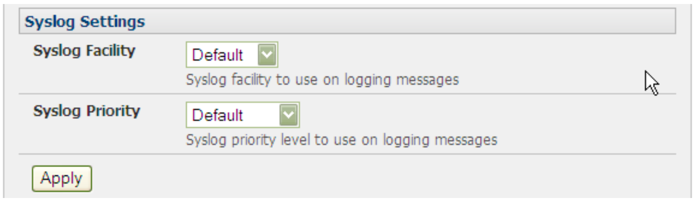


- You may secure the communications over the local Ethernet by enabling SSH however you will need to generate and upload keys (refer *Chapter 14 – Advanced Configuration*)

4.1.7 Syslog

In addition to inbuilt logging and monitoring (which can be applied to serial-attached and network-attached management accesses, as covered in *Chapter 7 - Alerts and Logging*) the Console Server can also be configured to support the remote syslog protocol on a per serial port basis:

- Select the **Syslog Facility/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 (i.e. redirect them/ send alert email etc.)



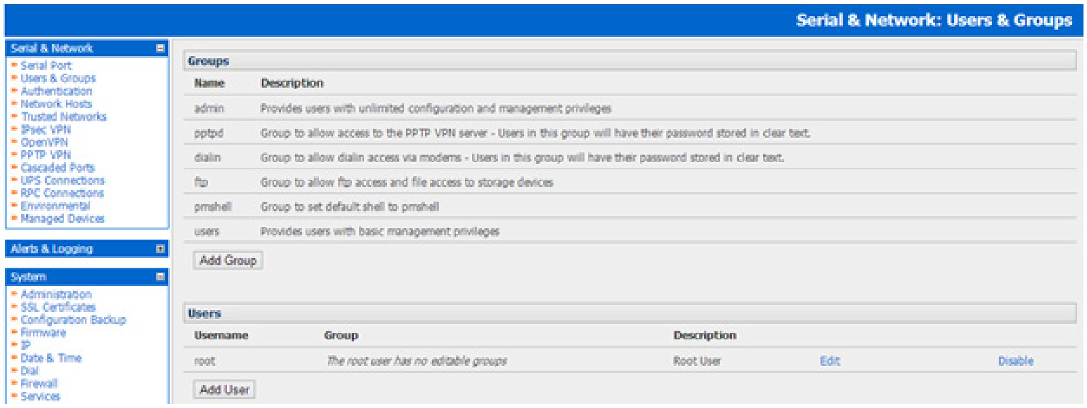
Syslog Settings

Syslog Facility	Default Syslog facility to use on logging messages
Syslog Priority	Default Syslog priority level to use on logging messages

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 **Facility** for that port to *local0* (*local0* .. *local7* are meant for site local values), and the **Priority** to *critical*. At this priority, if the Console Server syslog server does receive a message, it will automatically raise an alert. Refer to *Chapter 7 - Alerts & Logging*.

4.2 Add/ 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.



Groups

Name	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 dialin access via modems - 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

Users

Username	Group	Description		
root	*The root user has no editable groups*	Root User	Edit	Disable

Users can be authorized to access specified Console Server serial ports 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. There are two Groups set up by default (*admin* and *user*)

1. 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 which 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. However once logged in they can reconfigure the Console Server settings (e.g. to enable 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.

Note*: For convenience the SDT Connector "Retrieve Hosts" function retrieves and auto-configures checked serial ports and checked hosts only, even for admin group users*

2. 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.

3. With firmware V3.8.1 and later, there are six Groups set up by default (where earlier versions only had admin and user by default):

   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 dialin access via modems. 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

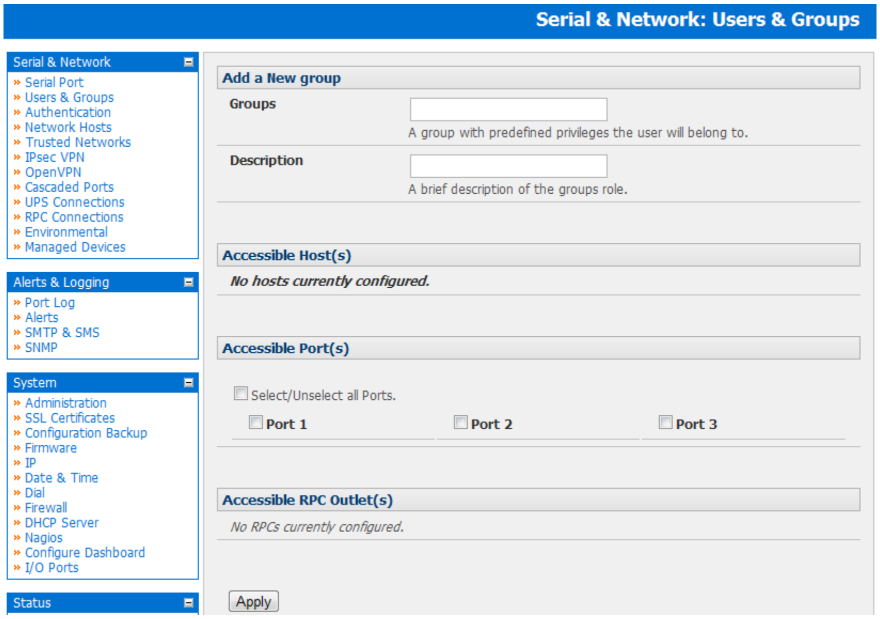
   users Provides users with basic management privileges

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

4. The Administrator can also set up additional Groups with specific serial port and host access permissions (same as Users). However 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. Lastly the Administrator can also set up users who are not a member of any Groups and they will have the same access as users in the additional 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



Add a New group

Groups

A group with predefined privileges the user will belong to.

Description

A brief description of the groups role.

Accessible Host(s)

No hosts currently configured.

Accessible Port(s)

□ Select/Unselect all Ports.

□ **Port 1** □ **Port 2** □ **Port 3**

Accessible RPC Outlet(s)

No RPCs currently configured.

- Add a **Group** name and **Description** for each new Group, then nominate the **Accessible Hosts, Accessible Ports** and **Accessible RPC Outlets(s)** that you wish any users in this new Group to be able to access
- Click **Apply**
- Click **Add User** to add a new user
- Add a **Username** and a confirmed **Password** 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 (however you can also use the special characters "-" "_" and "."). There are no restrictions on the characters that can be used in the user Password (which each can contain up to 254 characters). However, only the first eight Password characters are used to make the* password *hash.*

- Specify which **Group** (or Groups) you wish the user to be a member of
- Check specific **Accessible Hosts** and/or **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 (i.e. Power On/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 the number of users you can set up; nor on the number of users per serial port or host. So multiple users (Users and Administrators) can control /monitor the one port or host. Similarly there are no specific limits on the number of Groups and each user can be a member of a number of Groups (in which case they take on the cumulative access privileges of each of those Groups). 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 they will not be able to use the Management Console to manage ports).*

However while there are no specific limits the time to re-configure does increase as the number and complexity increases so we recommend the aggregate number of users and groups be kept under 250 (1000 for B092-016)

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

- Select Serial & Network: Users & Groups and click Edit for the User to be modified

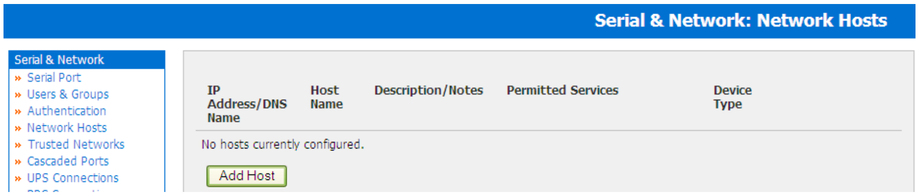
Note*: For more information on enabling the SDT Connector so each user has secure tunneled remote RPD/VNC/Telnet/HHTP/HTTPS/SoL access to the network connected hosts refer to* ***Chapter 6.***

4.3 Authentication

Refer to *Chapter 9.1 - Remote Authentication Configuration* for authentication configuration details

4.4 Network Hosts

To access a locally networked computer or device (referred to as a Host) you must identify the Host and specify the TCP or UDP ports/services that will be used to control that Host:



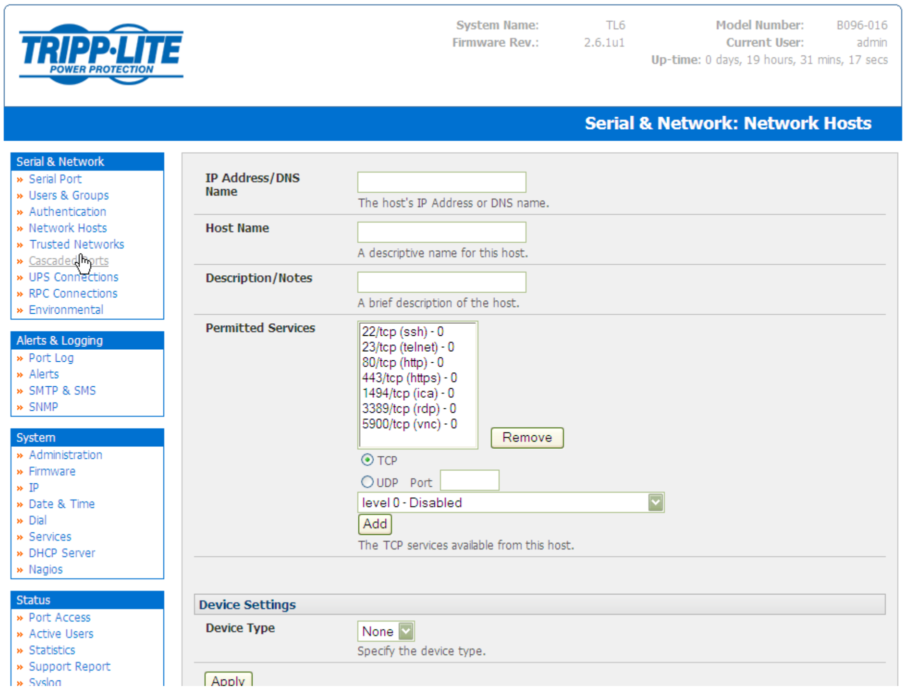
Serial & Network: Network Hosts

IP Address/DNS Name	Host Name	Description/Notes	Permitted Services	Device Type

No hosts currently configured.

- Selecting **Serial & Network: Network Hosts** presents all the network connected Hosts that have been enabled for access, and the related access TCP ports/services

- Click **Add Host** to enable access to a new Host (or select **Edit** to update the settings for existing Host)



System Name: TL6
Firmware Rev.: 2.6.1u1
Model Number: B096-016
Current User: admin
Up-time: 0 days, 19 hours, 31 mins, 17 secs

Serial & Network: Network Hosts

IP Address/DNS Name: The host's IP Address or DNS name.

Host Name: A descriptive name for this host.

Description/Notes: A brief description of the host.

Permitted Services:
22/tcp (ssh) - 0
23/tcp (telnet) - 0
80/tcp (http) - 0
443/tcp (https) - 0
1494/tcp (ica) - 0
3389/tcp (rdp) - 0
5900/tcp (vnc) - 0

● TCP
○ UDP Port
level 0 - Disabled

The TCP services available from this host.

Device Settings

Device Type: None
Specify the device type.

- Enter the **IP Address** or **DNS Name** and a **Host Name** (up to 254 alphanumeric characters) for the new network connected Host (and optionally enter a **Description** -up to characters)

- 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.

- The **Logging Level** specifies the level of information to be logged and monitored for each Host access (refer *Chapter 7 - Alerts and Logging*)

- If the Host is a networked server with IPMI power control, then specify **RPC** (for IPMI and PDU) or **UPS** and the **Device Type**. The Administrator can then configure these devices and enable which users have permissions to remotely cycle power etc (refer *Chapter 8*). Otherwise leave the Device Type set to None

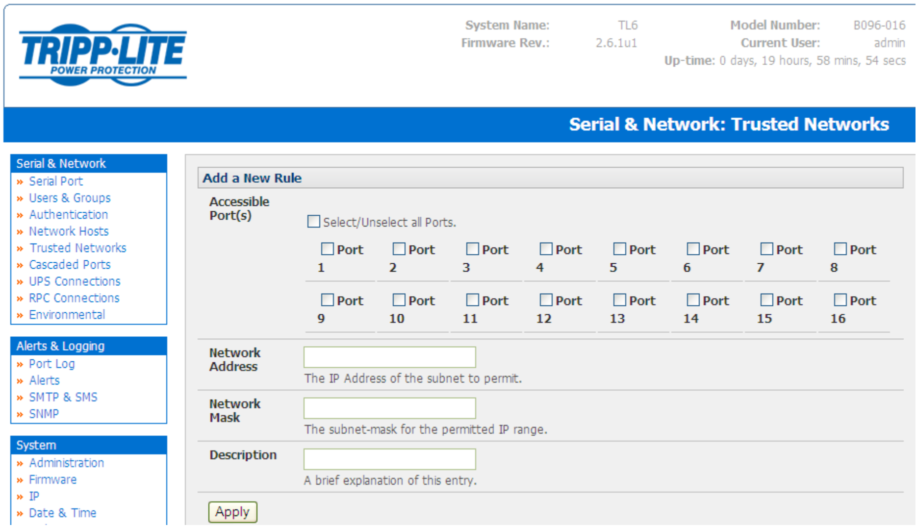
- If the Console Server has been configured with distributed Nagios monitoring enabled then you will also be presented with **Nagios Settings** options to enable nominated services on the Host to be monitored (refer *Chapter 10 – Nagios Integration*)

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

4.5 Trusted Networks

The **Trusted Networks** facility gives you an option to nominate specific IP addresses that users (Administrators and Users) must be located at, to have access to Console Server serial ports:

- Select **Serial & Network: Trusted Networks**
- To add a new trusted network, select **Add Rule**



System Name: TL6
Firmware Rev.: 2.6.1u1
Model Number: B096-016
Current User: admin
Up-time: 0 days, 19 hours, 58 mins, 54 secs

Serial & Network: Trusted Networks

Add a New Rule

Accessible Port(s)

□ Select/Unselect all Ports.

□ Port 1 □ Port 2 □ Port 3 □ Port 4 □ Port 5 □ Port 6 □ Port 7 □ Port 8

□ Port 9 □ Port 10 □ Port 11 □ Port 12 □ Port 13 □ Port 14 □ Port 15 □ Port 16

Network Address: The IP Address of the subnet to permit.

Network Mask: The subnet-mask for the permitted IP range.

Description: A brief explanation of this entry.

- Select the **Accessible Port(s)** that the new rule is to be applied to
- Then enter the **Network Address** of the subnet to be permitted access
- Then specify the range of addresses that are to be permitted by entering a **Network Mask** for that permitted IP range e.g.
  - To permit all the users located with a particular Class C network (204.15.5.0 say) connection to the nominated port then you would add the following Trusted Network New Rule:

Network IP Address	204.15.5.0
Subnet Mask	255.255.255.0

  - If you want to permit only the one users who is located at a specific IP address (204.15.5.13 say) to connect:

Network IP Address	204.15.5.0
Subnet Mask	255.255.255.255

  - If however you want to allow all the users operating from within a specific range of IP addresses (say any of the thirty addresses from 204.15.5.129 to 204.15.5.158) to be permitted connection to the nominated port:

Host /Subnet 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. However 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 the **Chapter 14 - Advanced.***

4.6 Serial Port Cascading

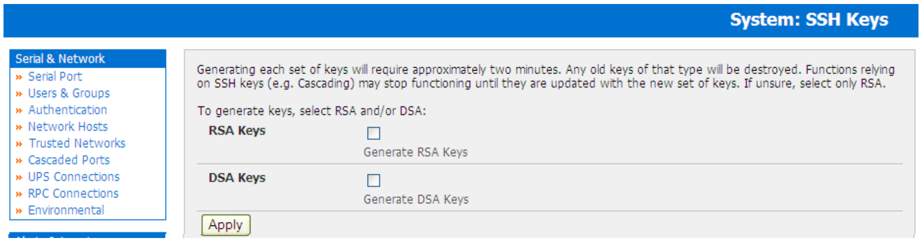
Cascaded Ports enables you to cluster distributed Console Servers so that a large number of serial ports (up to 1000) can be configured and accessed through one IP address and managed through the one Management Console. One Console Server, the Master, controls other Console Servers as Slave units and all the serial ports on the Slave units appear as if they are part of the Master.

Each Slave connects to the Master with an SSH connection using public key authentication. So the Master accesses each Slave using an SSH key pair, rather than using passwords, ensuring secure authenticated communications. So the Slave Console Server units can be distributed locally on a LAN or remotely over public networks around the world.

4.6.1 Automatically generate and upload SSH keys

To set up public key authentication you must first generate an RSA or DSA key pair and upload them into the Master and Slave Console Servers. This can all be done automatically from the Master:

- Select **System: Administration** on Master's Management Console
- Check **Generate SSH keys automatically** and click **Apply**



Generating each set of keys will require approximately two minutes. Any old keys of that type will be destroyed. Functions relying on SSH keys (e.g. Cascading) may stop functioning until they are updated with the new set of keys. If unsure, select only RSA.

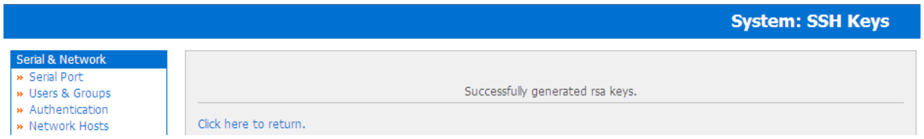
To generate keys, select RSA and/or DSA:

RSA Keys ☐ Generate RSA Keys

DSA Keys ☐ Generate DSA Keys

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

- Select **RSA Keys** and/or **DSA Keys**
- Click **Apply**



Successfully generated rsa keys.

Click here to return.

- Once the new keys have been successfully generated simply Click **here to return** and the keys will automatically be uploaded to the Master and connected Slaves

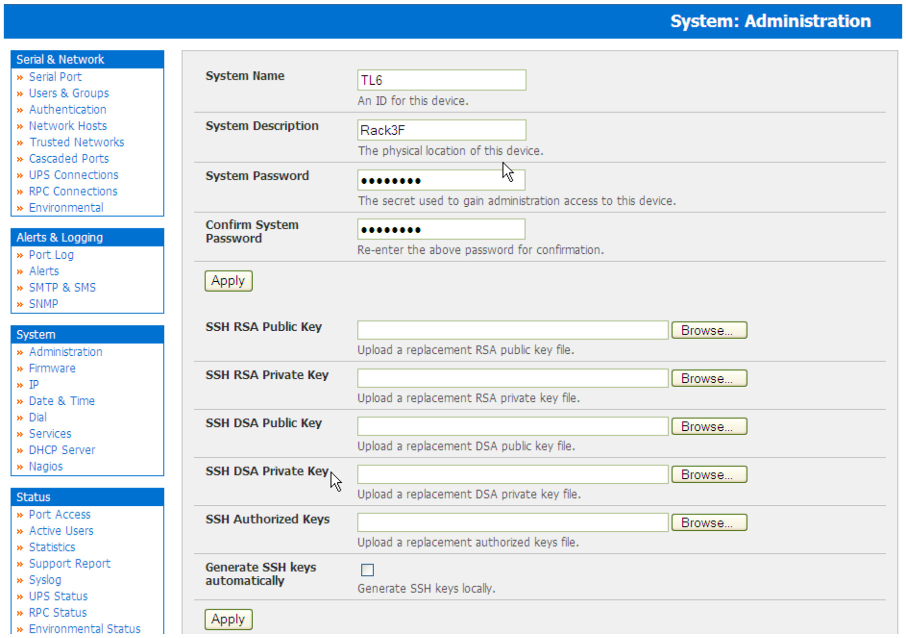
4.6.2 Manually generate and upload SSH keys

Alternately if you have a RSA or DSA key pair you can manually upload them to the Master and Slave Console Servers.

Note*: If you do not already have RSA or DSA key pair and you do not wish to use you will need to create a key pair using* ***ssh-keygen****,* ***PuTTYgen*** *or a similar tool as detailed in Chapter 15.6*

To manually upload the key public and private key pair to the Master Console Server:

- Select **System: Administration** on 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**



System Name: TL6
An ID for this device.

System Description: Rack3F
The physical location of this device.

System Password: ••••••••
The secret used to gain administration access to this device.

Confirm System Password: ••••••••
Re-enter the above password for confirmation.

SSH RSA Public Key
Upload a replacement RSA public key file.

SSH RSA Private Key
Upload a replacement RSA private key file.

SSH DSA Public Key
Upload a replacement DSA public key file.

SSH DSA Private Key
Upload a replacement DSA private key file.

SSH Authorized Keys
Upload a replacement authorized keys file.

Generate SSH keys automatically ☐
Generate SSH keys locally.

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 refer the* ***Authorized Keys*** *section of Chapter 15.6. 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
- 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:

- To establish the fingerprint first log in the Master server as root and establish an SSH connection to the Slave remote host:

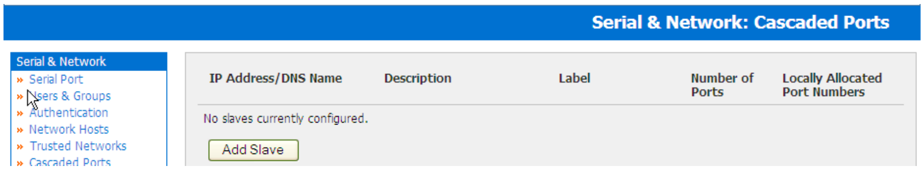
  # ssh remhost

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 refer Chapter 15.6

- If you are asked to supply a password, then there has been a problem with uploading keys. The keys should remove any need to supply a password

4.6.3 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:



IP Address/DNS Name	Description	Label	Number of Ports	Locally Allocated Port Numbers
No slaves currently configured.				

- Select **Serial & Network: Cascaded Ports** on the Master's Management Console:
- To add clustering support select **Add Slave**

***Note**: You will be prevented from adding 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 e.g.

- 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
- The configuration changes made on the Master are propagated out to all the Slaves when you click **Apply**.

4.6.4 Managing the slaves

The Master is in control of the Slave serial ports. So for example if 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). However these changes will be overwritten next time the Master sends out a configuration file update.

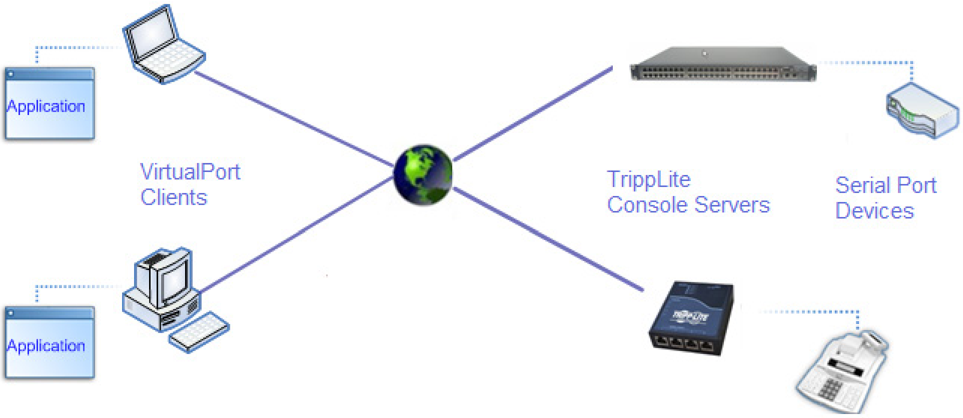
Also while the Master is in control of all Slave serial port related functions, it is not master over the Slave network host connections or over the Slave Console Server system itself.

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 over written when configuration changes are propagated from the Master. Similarly the Slaves Network Host and IPMI settings have to be configured at each Slave.

Also the Master's Management Console provides a consolidated view of the settings for its own and all the Slave's serial ports, however 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 11.

4.7 Serial Port Redirection

Tripp Lite's VirtualPort software delivers the virtual serial port technology your Windows applications need to open remote serial ports and read the data from serial devices that are connected to your Console Server.



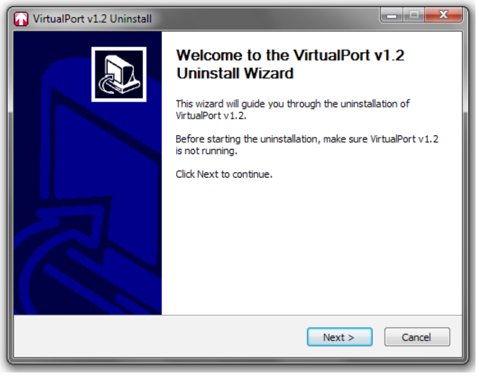
VirtualPort is supplied with each B096-016 / B096-032 / B096-048 Console Server Management Switch or B092-016 Console Server with PowerAlert or B095-003-1E-M / B095-004-1E Console Server.

You are licensed to install *VirtualPort* on one or more computers for accessing any serial device connected to any Tripp Lite Console Server port.

4.7.1 Install VirtualPort client

VirtualPort is fully compatible with 32-bit and 64-bit versions of Windows NT 4.x, Windows XP, Windows 2000, Windows 2003, Windows 2008, Windows Vista and 64-bit and Windows 7. The installation process is simple.

- The *virtualport_setup.exe* program is included on the CD supplied with your Console Server (or a copy can be freely downloaded from the ftp site.) Double click the *VirtualPort_setup.exe* file to start installation process



Welcome to the VirtualPort v1.2 Uninstall Wizard

This wizard will guide you through the uninstallation of VirtualPort v1.2.

Before starting the uninstallation, make sure VirtualPort v1.2 is not running.

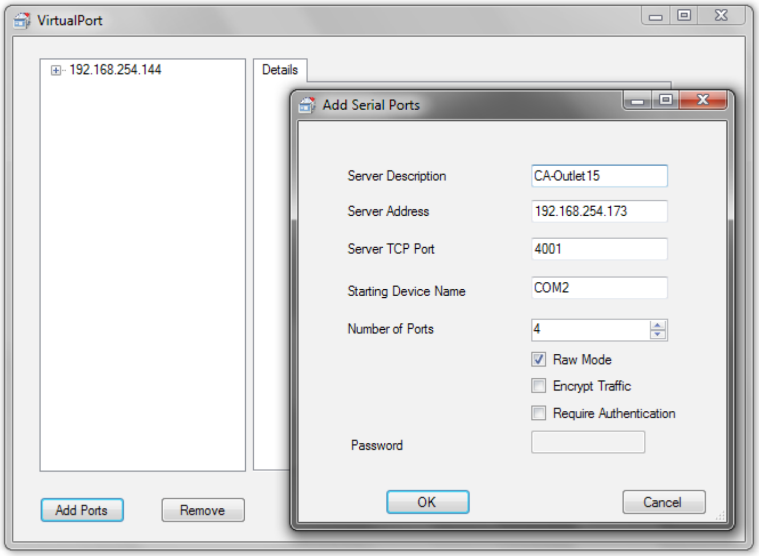
Click Next to continue.

- Read the License Agreement then follow the prompts to select the destination path and choose shortcuts you wish to create Once the installer completes you will have a working *VirtualPort* client installed on your machine and an icon on your desktop
- Click the *VirtualPort* icon on your desktop to start the client

4.7.2 Configure the VirtualPort client

Creating the VirtualPort client connection will initiate a virtual serial port data redirection to the remote Console Server using TCP/IP protocol

- Click on *Add Ports*
- Specify a name to identify this connection in the "Server Description " tab



Server Description: CA-Outlet15

Server Address: 192.168.254.173

Server TCP Port: 4001

Starting Device Name: COM2

Number of Ports: 4

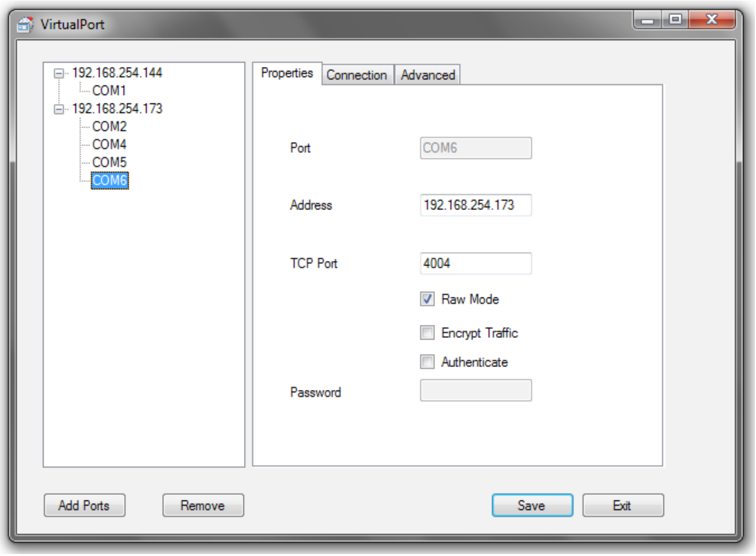
☑ Raw Mode

□ Encrypt Traffic

□ Require Authentication

Password:

- Enter the Console Server's IP address (or network name)
- Enter the *Server TCP Port* number that matches the port you have configured for the serial device on the remote Console Server. Ensure this port isn't blocked by firewall
  - o Telnet *RFC2217* mode is configured by default so the range of port numbers available on a 16 port console server would be 5001-5016
  - o Alternately check *RAW* mode (4001- 4048 on a 48 port console server)
  - o Select *Encrypted* to enable SSL/TLS encryption of the data going to the port. You will need to enter a *Password*
- Select the starting COM port (COM1 to COM4096)
- Specify the number of ports you want to add. Sequential port numbers will be assigned automatically however if a COM port # is already being used by other applications that # will be skipped
- Click **OK** to add the specified COM ports



Port COM6

Address 192.168.254.173

TCP Port 4004

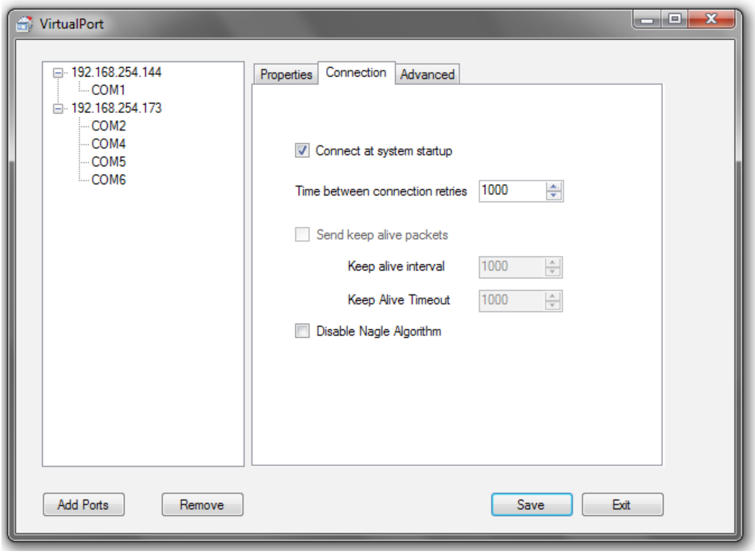
☑ Raw Mode

□ Encrypt Traffic

□ Authenticate

Password

- To configure a COM port you have created simply click on the desired COMx label in the left hand menu tree
- In the Properties window you can edit the IP Address or TCP Port to be used to connect to that COM port
- You can then configure the COM port in the Connection and Advanced windows:



☑ Connect at system startup

Time between connection retries 1000

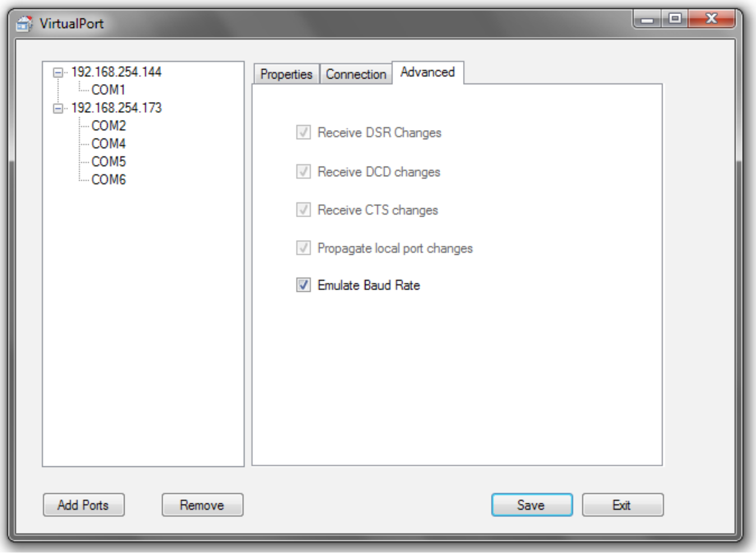
□ Send keep alive packets

Keep alive interval 1000

Keep Alive Timeout 1000

□ Disable Nagle Algorithm

- *Connect at system startup*—When enabled *VirtualPort* will try to connect to the Console Server when the *VirtualPort* service starts (as opposed to waiting for the application to open the serial port before initiating the connection to the Console Server)
- The *Time between connection retries* specifies the number of seconds between TCP connection retries after a client-initiated connection failure. Valid values are 1-255 (The default is 1 second and *VirtualPort* will continue attempting to reconnect forever to the Console Server at this interval)
- The *Send keep alive packets* option tests if the TCP connection is still up when no data has been sent for a while by sending keep-alive messages. Select this option and specify period of time (in milliseconds) after which *VirtualPort* sends a command to remote Console Server end in order to verify connection's integrity and keep the connection alive
- The *Keep Alive Interval* specifies the number of seconds to wait on an idle connection before sending a keep-alive message. The default is 1 second. The Keep Alive Timeout specifies how long VirtualPort should wait for a keep alive response before timing out the connection.
- *Disable Nagle Algorithm* — the Nagle Algorithm is enabled by default and it reduces the number of small packets sent by *VirtualPort* across the network



☑ Receive DSR Changes

☑ Receive DCD changes

☑ Receive CTS changes

☑ Propagate local port changes

☑ Emulate Baud Rate

- Check *Receive DSR/DCD/CTS changes* if the flow control signal status from the physical serial port on Console Server is to be reflected back to the Windows COM port driver (as some serial communications applications prefer to run without any hardware flow control i.e. in “two wire” mode)
- The *Propagate local port changes* allows complete serial device control by the Windows application so it operates exactly like a directly connected serial COM port. It provides a complete COM port interface between the attached serial device and the network, providing hardware and software flow control. So the baud rate of the remote serial port is controlled by the settings for that COM port on Windows computer. If not selected then the port serial configuration parameters are set on the **Console Server.**
- With the *Emulate Baud Rate* selected *VirtualPort* will only send data out at the baud rate configured by the local Application using the COM port

4.7.3 To remove a configured port

At any stage you can delete a single configured COM port, or delete the **Console Server** connection (and all the COM ports configured on that Console Server)

- Select the console server or COM port on the left hand menu and click the *Remove* button

4.7.4 Configure the remote serial device connection

Ensure the remote serial device is connected to your remote **Console Server**. Then configure the serial port as detailed in the User Guide

- Set the RS232 Common Settings (e.g. baud rate)
- Select Console server mode and specify the appropriate protocol to be used:
  - o *RAW TCP* allows connections directly to a TCP socket and the default TCP port address is 4000 + serial port # (i.e. the address of the second serial port is *IP Address _ 4002*)
  - o *RFC2217* enables serial port redirection on that port and the default port address is IP Address _ Port (5000 + serial port #) i.e. 5001 – 5048 on a 48 port **Console Server**

4.8 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 will display all the Managed Device with their Description/Notes and lists of all the configured Connections:

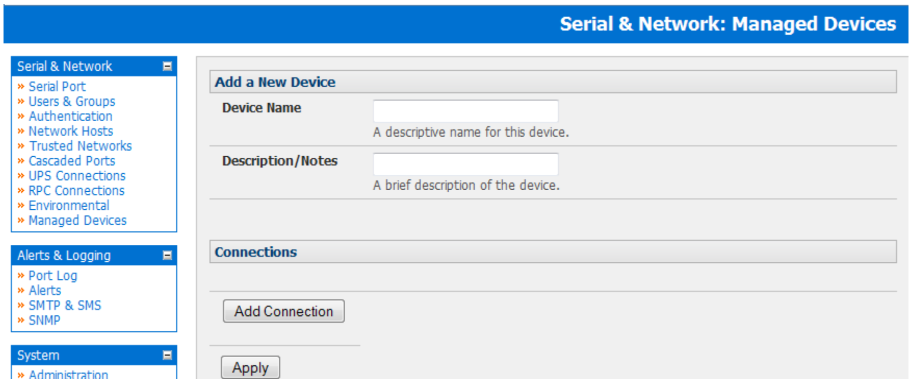
- *Serial Port #* (if serially connected) or
- *USB* (if USB connected)
- *IP Address* (if network connected)
- *Power PDU*/outlet details (if applicable) and any UPS connections

Devices such as servers will commonly have more than one power connections (e.g. dual power supplied) and more than one network connection (e.g. for BMC/service processor).

All users can view (but not edit) these Managed Device connections by selecting Manage: Devices. The Administrator can edit and add/delete these Managed Devices and their connections.

To edit an existing device and add a new connection:

- Select **Edit** on the **Serial&Network: Managed Devices** and click **Add Connection**
- Select the connection type for the new connection (Serial, Network Host, UPS or RPC) and then select the specific connection from the presented list of configured unallocated hosts/ports/outlets



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 4.4 - Network Hosts*)
- 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. Again corresponding new Managed Device (with the same Name /Description as the RPC/UPS Host) is not created until this connection step is completed (refer *Chapter 8 - Power and Environment*)

To add a new serially connected Managed Device:

- Configure the serial port using the **Serial&Network: Serial Port** menu (refer *Section 4.1 -Configure Serial Port*)
- Select **Serial&Network: Managed Devices** and 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
- To add a UPS/RPC power connection or network connection or another serial connection click **Add 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 / Description as the RPC/UPS Host (refer* Chapter 8 - Power and Environment*)*

Also all the outlet names on the PDU will by default be "Outlet 1" "Outlet 2". When you connect an particular Managed Device (that draws power from the outlet) they the outlet will then take up the name of the powered Managed Device

4.9 IPsec VPN

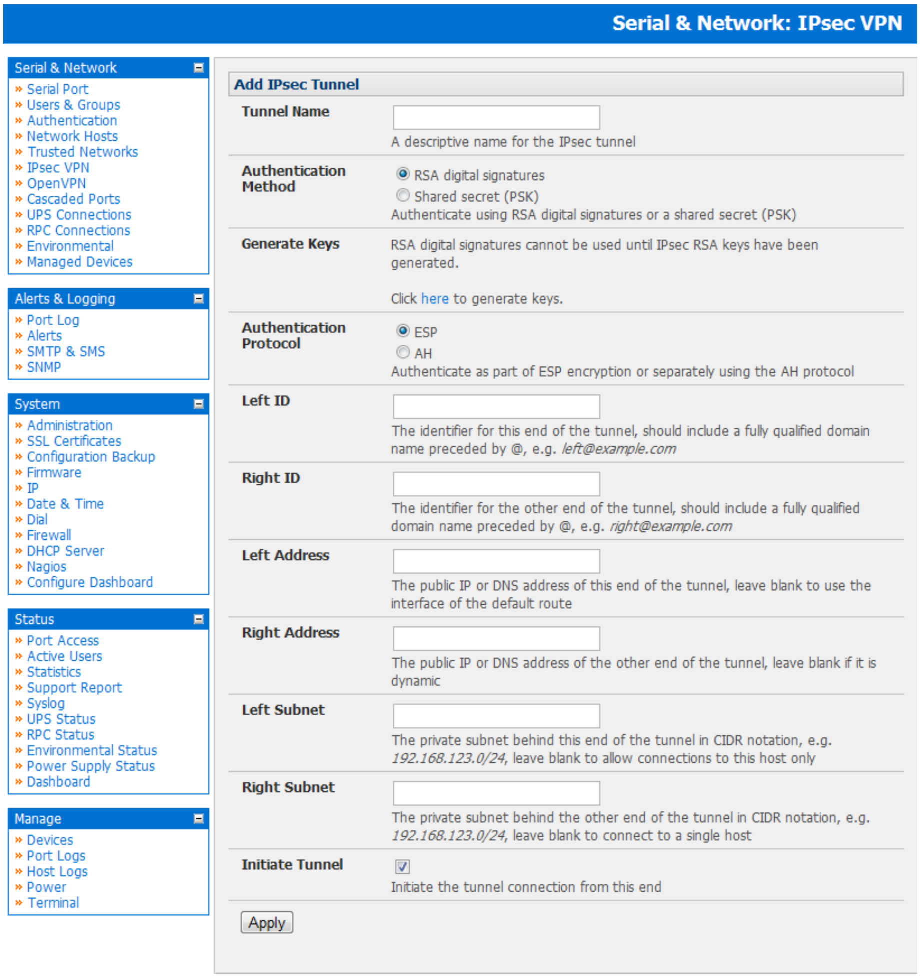
The Console Servers include Openswan, a Linux implementation of the IPsec (IP Security) protocols, which can be used to configure a Virtual Private Network (VPN). The VPN allows multiple sites or remote administrators to access the Console Server (and Managed Devices) securely over the Internet.

- The administrator can establish an encrypted authenticated VPN connections between Console Servers distributed at remote sites and a VPN gateway (such as Cisco router running *IOS IPsec*) on their central office network:
    - o 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
    - o 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 (refer Chapter 4.1)
- The road warrior administrator can use a VPN IPsec software client such as TheGreenBow (www.thegreenbow.com/vpn_gateway.html) or Shrew Soft (www.shrew.net/support) to remotely access the Console Server and every machine on the Management LAN subnet at the remote location

Configuration of IPsec is quite complex so Tripp Lite provides a simple GUI interface for basic set up as described below. However for more detailed information on configuring Openswan IPsec at the command line and interconnecting with other IPsec VPN gateways and road warrior IPsec software refer **http://wiki.openswan.org**

4.9.1 Enable the VPN gateway

- Select **IPsec VPN** on the **Serial & Networks** menu
- Click **Add** and complete the *Add IPsec Tunnel* screen
- Enter any descriptive name you wish to identify the IPsec Tunnel you are adding such as *WestStOutlet-VPN*



Add IPsec Tunnel

Tunnel Name
A descriptive name for the IPsec tunnel

Authentication Method
● RSA digital signatures
○ Shared secret (PSK)
Authenticate using RSA digital signatures or a shared secret (PSK)

Generate Keys
RSA digital signatures cannot be used until IPsec RSA keys have been generated.

Click here to generate keys.

Authentication Protocol
● ESP
○ AH
Authenticate as part of ESP encryption or separately using the AH protocol

Left ID
The identifier for this end of the tunnel, should include a fully qualified domain name preceded by @, e.g. *left@example.com*

Right ID
The identifier for the other end of the tunnel, should include a fully qualified domain name preceded by @, e.g. *right@example.com*

Left Address
The public IP or DNS address of this end of the tunnel, leave blank to use the interface of the default route

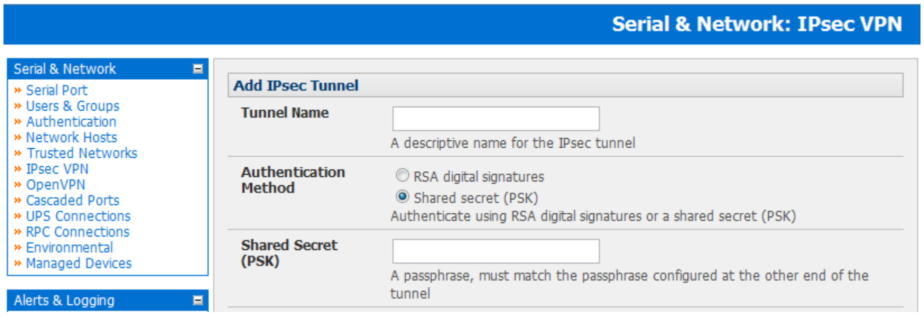
Right Address
The public IP or DNS address of the other end of the tunnel, leave blank if it is dynamic

Left Subnet
The private subnet behind this end of the tunnel in CIDR notation, e.g. *192.168.123.0/24*, leave blank to allow connections to this host only

Right Subnet
The private subnet behind the other end of the tunnel in CIDR notation, e.g. *192.168.123.0/24*, leave blank to connect to a single host

Initiate Tunnel
☑
Initiate the tunnel connection from this end

- Select the **Authentication Method** to be used, either *RSA digital signatures* or a *Shared secret (PSK)*
    - o 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*



Add IPsec Tunnel

Tunnel Name
A descriptive name for the IPsec tunnel

Authentication Method
○ RSA digital signatures
● Shared secret (PSK)
Authenticate using RSA digital signatures or a shared secret (PSK)

Shared Secret (PSK)
A passphrase, must match the passphrase configured at the other end of the tunnel

- o 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 '@' (e.g. *left@example.com*)
- Enter the public IP or DNS address of the gateway device connecting it to the Internet as the **Left Address**. You can leave this blank to use the interface of the default route

- In **Right Address** enter the public IP or DNS address of the remote end of the tunnel (only if the remote end has a static or dyndns address). Otherwise leave this blank
- If the VPN gateway is serving 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 the 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 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.*

4.10 OpenVPN

Console Servers also include OpenVPN which is based on TSL (Transport Layer Security) and SSL (Secure Socket Layer). With OpenVPN, it is easy to build cross-platform, point-to-point VPNs using x509 PKI (Public Key Infrastructure) or custom configuration files.

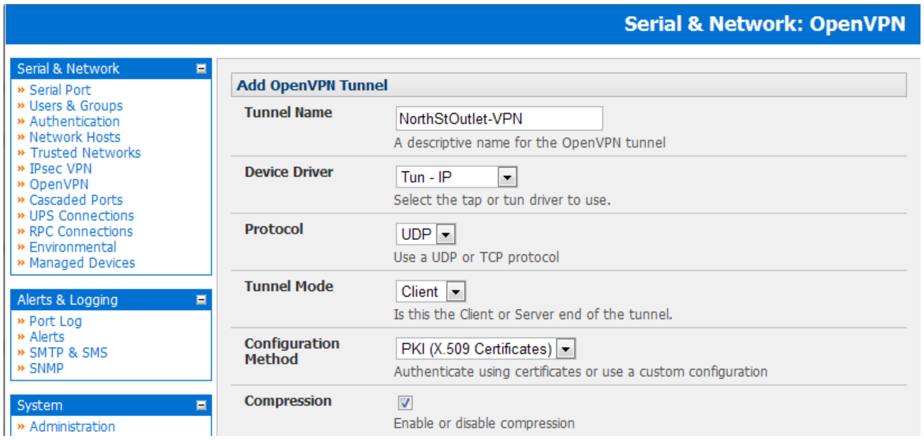
OpenVPN allows secure tunneling of data through a single TCP/UDP port over an unsecured network, thus providing secure access to multiple sites and secure remote administration to a console server over the Internet.

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 a Console Server within a data centre.

Configuration of OpenVPN can be complex so Tripp Lite provides a simple GUI interface for basic set up as described below. However for more detailed information on configuring OpenVPN Access server or client refer to the HOW TO and FAQs at **http://www.openvpn.net**

4.10.1 Enable the OpenVPN

- Select **OpenVPN** on the **Serial & Networks** menu
- Click **Add** and complete the *Add OpenVPN Tunnel* screen
- Enter any descriptive name you wish to identify the OpenVPN Tunnel you are adding, for example *NorthStOutlet-VPN*



Add OpenVPN Tunnel

Tunnel Name	NorthStOutlet-VPN A descriptive name for the OpenVPN tunnel
Device Driver	Tun - IP Select the tap or tun driver to use.
Protocol	UDP Use a UDP or TCP protocol
Tunnel Mode	Client Is this the Client or Server end of the tunnel.
Configuration Method	PKI (X.509 Certificates) Authenticate using certificates or use a custom configuration
Compression	☑ Enable or disable compression

- 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 Console Server supports multiple clients connecting to the VPN server over the same port.
- In Configuration Method, select the authentication method to be used. To authenticate using certificates select PKI (X.509 Certificates) or select Custom Configuration to upload custom configuration files. Custom configurations must be stored in /etc/config.

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

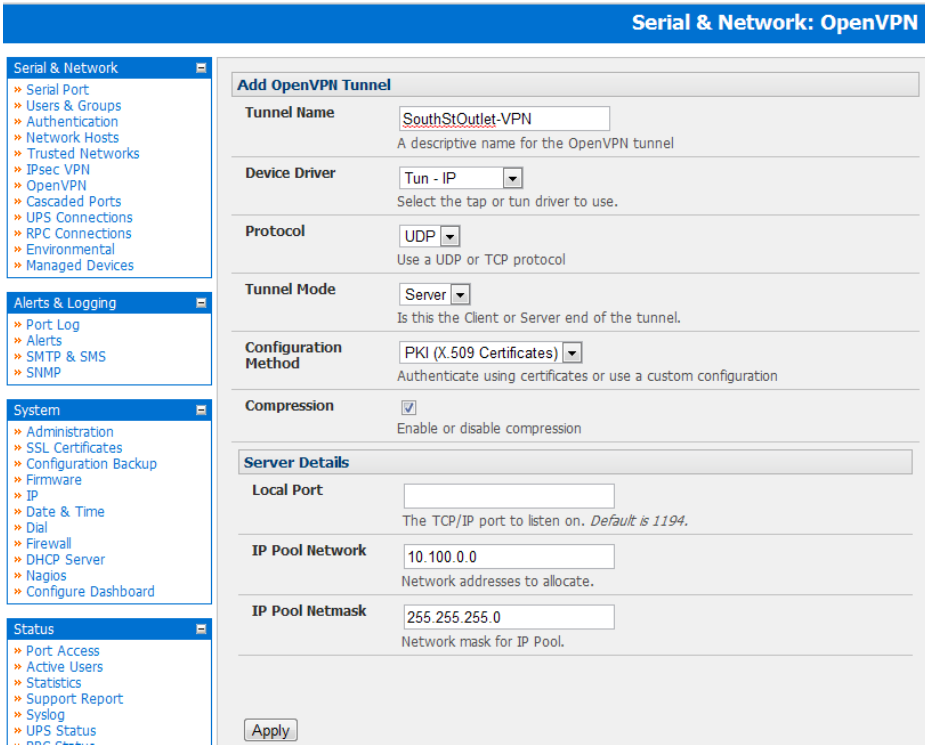
- *Separate certificate (also known as a public key). This **Certificate File** will be a ***.crt** file type*
- *Private Key for the server and each client. This **Private Key File** will be a ***.key** file type*
- *Master Certificate Authority (CA) certificate and key which is used to sign each of 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).*
*Refer **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***

- Check or uncheck the **Compression** button to enable or disable compression, respectively

4.10.2 Configure as Server or Client

- Complete the **Client Details** or **Server Details** depending on the Tunnel Mode selected.
    - o If *Client* has been selected, the *Primary Server Address* will be the address of the OpenVPN Server.
    - o If *Server* has been selected, enter the IP Pool Network address and the IP Pool Network mask for the IP Pool. The network defined by the IP Pool Network address/mask is *used to provide the addresses for connecting clients.*
- Click **Apply** to save changes



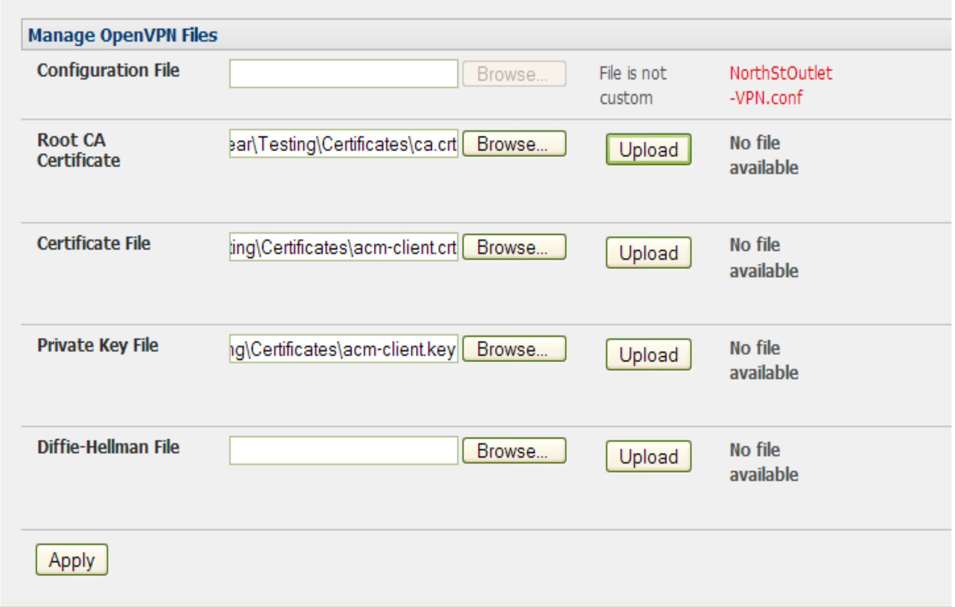
Add OpenVPN Tunnel

Tunnel Name	SouthStOutlet-VPN A descriptive name for the OpenVPN tunnel
Device Driver	Tun - IP Select the tap or tun driver to use.
Protocol	UDP Use a UDP or TCP protocol
Tunnel Mode	Server Is this the Client or Server end of the tunnel.
Configuration Method	PKI (X.509 Certificates) Authenticate using certificates or use a custom configuration
Compression	☑ Enable or disable compression

Server Details

Local Port	 The TCP/IP port to listen on. *Default is 1194.*
IP Pool Network	10.100.0.0 Network addresses to allocate.
IP Pool Netmask	255.255.255.0 Network mask for IP Pool.

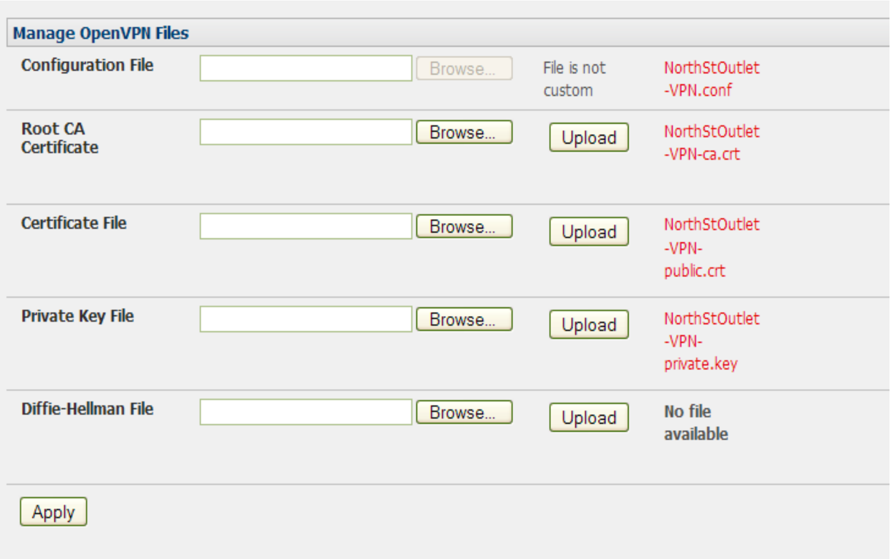
- To enter authentication certificates and files **Edit** the OpenVPN tunnel.
- Select the **Manage OpenVPN Files** tab. Upload or browse to relevant authentication certificates and files.



Manage OpenVPN Files

Configuration File		Browse...	File is not custom	NorthStOutlet-VPN.conf
Root CA Certificate	ear\Testing\Certificates\ca.crt	Browse...	Upload	No file available
Certificate File	ing\Certificates\acm-client.crt	Browse...	Upload	No file available
Private Key File	ng\Certificates\acm-client.key	Browse...	Upload	No file available
Diffie-Hellman File		Browse...	Upload	No file available

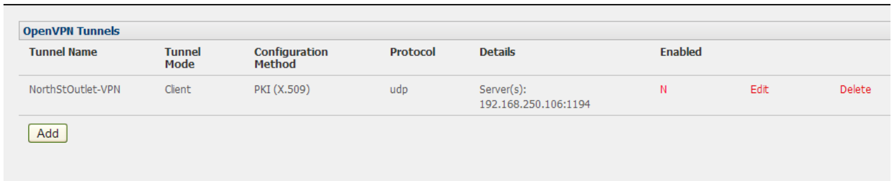
- **Apply** to save changes. Saved files will be displayed in red on the right-hand side of the Upload button.



Manage OpenVPN Files

Configuration File	Browse...	File is not custom	NorthStOutlet-VPN.conf
Root CA Certificate	Browse...	Upload	NorthStOutlet-VPN-ca.crt
Certificate File	Browse...	Upload	NorthStOutlet-VPN-public.crt
Private Key File	Browse...	Upload	NorthStOutlet-VPN-private.key
Diffie-Hellman File	Browse...	Upload	No file available

- To enable OpenVPN, **Edit** the OpenVPN tunnel

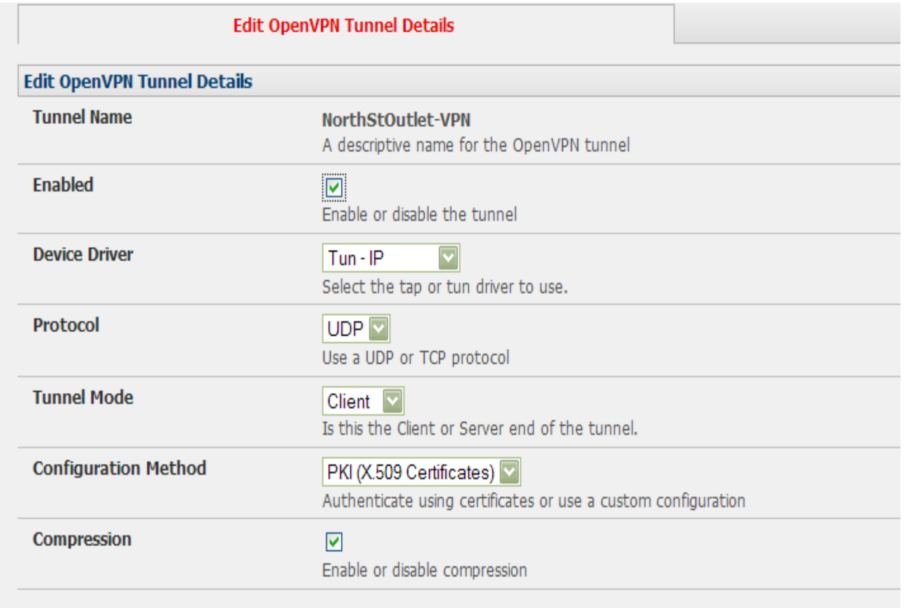


OpenVPN Tunnels

Tunnel Name	Tunnel Mode	Configuration Method	Protocol	Details	Enabled		
NorthStOutlet-VPN	Client	PKI (X.509)	udp	Server(s): 192.168.250.106:1194	N	Edit	Delete

- Check the **Enabled** button.
- **Apply** to save changes

***Note**: Please make sure that the console server system time is correct when working with OpenVPN. Otherwise authentication issues may arise*



Edit OpenVPN Tunnel Details

Tunnel Name	NorthStOutlet-VPN A descriptive name for the OpenVPN tunnel
Enabled	☑ Enable or disable the tunnel
Device Driver	Tun - IP Select the tap or tun driver to use.
Protocol	UDP Use a UDP or TCP protocol
Tunnel Mode	Client Is this the Client or Server end of the tunnel.
Configuration Method	PKI (X.509 Certificates) Authenticate using certificates or use a custom configuration
Compression	☑ Enable or disable compression

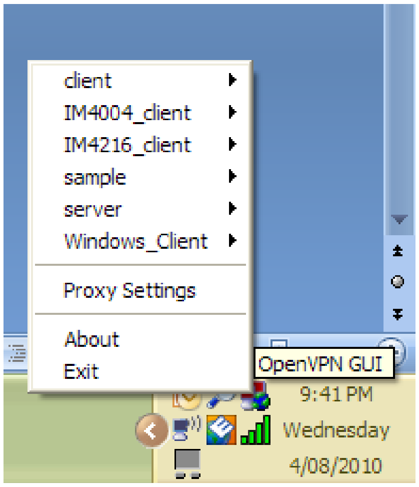
- Select **Statistics** on the **Status** menu to verify that the tunnel is operational.

4.10.3 Windows OpenVPN Client and Server set up

Windows does not come with an 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.

The OpenVPN GUI for Windows software (which includes the standard OpenVPN package plus a Windows GUI) can be downloaded from http://openvpn.se/download.html.

- Once installed on the Windows machine, an OpenVPN icon will have been created 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 will be rechecked for new configuration files whenever the OpenVPN GUI icon is right-clicked. So once OpenVPN 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 of an OpenVPN Windows client configuration file is shown below:

```
# description: BL_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 of an OpenVPN Windows Server configuration file is shown below:

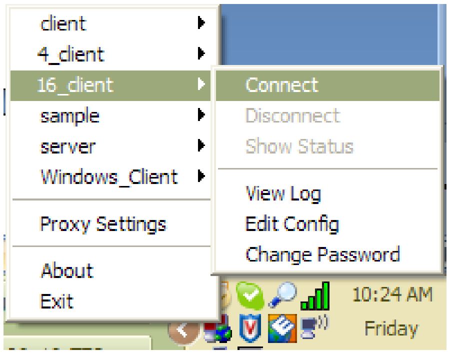
```
server 10.100.10.0 255.255.255.0
port 1194
keepalive 10 120
proto udp
mssfix 1400
persist-key
persist-tun
dev tun
ca c:\\openvpnkeys\\ca.crt
cert c:\\openvpnkeys\\server.crt
key c:\\openvpnkeys\\server.key
dh c:\\openvpnkeys\\dh.pem
comp-lzo
verb 1
syslog BL_OpenVPN_Server
```

The Windows client/server configuration file options are:

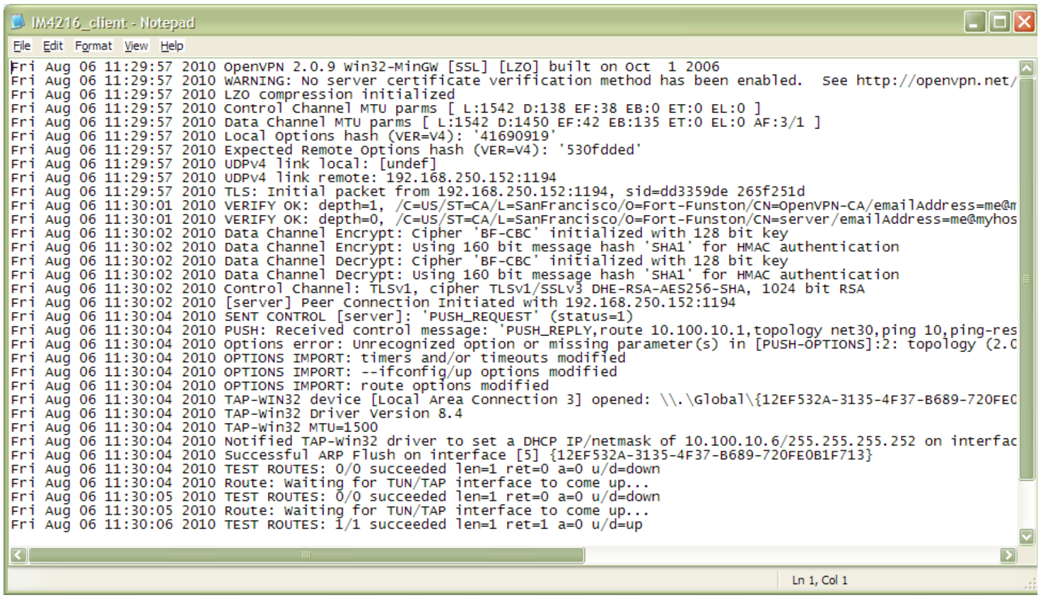
Options	Description
#description:	This is a comment describing the configuration. Comment lines start with a '#' and are ignored by OpenVPN.
Client server	Specify whether this will be a client or server configuration file. In the server configuration file, define the IP address pool and netmask. For example, server 10.100.10.0 255.255.255.0
proto udp proto tcp	Set the protocol to UDP or TCP. The client and server must use the same settings.
mssfix <max. size>	Mssfix sets the maximum size of the packet. This is only useful for UDP if problems occur.
verb <level>	Set log file verbosity level. Log verbosity level can be set from 0 (minimum) to 15 (maximum). For example, 0 = silent except for fatal errors 3 = medium output, good for general usage 5 = helps with debugging connection problems 9 = extremely verbose, excellent for troubleshooting
dev tun dev tap	Select 'dev tun' to create a routed IP tunnel or 'dev tap' to create an Ethernet tunnel. The client and server must use the same settings.
remote <host>	The hostname/IP of OpenVPN server when operating as a client. Enter either the DNS hostname or the static IP address of the server.
Port	The UDP/TCP port of the server.
Keepalive	Keepalive uses ping to keep the OpenVPN session alive. 'Keepalive 10 120' pings every 10 seconds and assumes the remote peer is down if no ping has been received over a 120 second time period.
http-proxy <proxy server> <proxy port #>	If a proxy is required to access the server, enter the proxy server DNS name or IP and port number.
ca <file name>	Enter the CA certificate file name and location. The same CA certificate file can be used by the server and all clients. Note: Ensure each '\' in the directory path is replaced with ' \\'. For example, c:\openvpnkeys\ca.crt will become c:\\openvpnkeys\\ca.crt
cert <file name>	Enter the client's or servers's certificate file name and location. Each client should have its own certificate and key files.Note: Ensure each '\' in the directory path is replaced with ' \\'.
key <file name>	Enter the file name and location of the client's or server's key. Each client should have its own certificate and key files. Note: Ensure each '\' in the directory path is replaced with ' \\'.
dh <file name>	This is used by the server only. Enter the path to the key with the Diffie-Hellman parameters.
Nobind	'Nobind' is 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	This option prevents the reloading of keys across restarts.
persist-tun	This option prevents the close and reopen of TUN/TAP devices across restarts.
cipher BF-CBC Blowfish (default) cipher AES-128-CBC AES cipher DES-EDE3-CBC Triple-DES	Select a cryptographic cipher. The client and server must use the same settings.
comp-lzo	Enable compression on the OpenVPN link. This must be enabled on both the client and the server.
syslog	By default, logs are located in syslog or, if running as a service on Window, in \Program Files\OpenVPN\log directory.

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. For example, BL_client
- Click 'Connect' as shown below

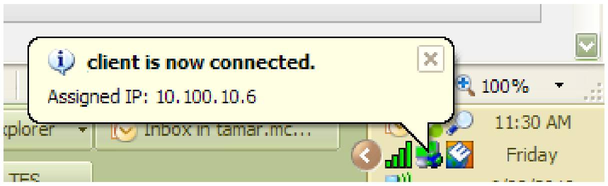


- The log file will be displayed as the connection is established



```
Fri Aug 06 11:29:57 2010 OpenVPN 2.0.9 Win32-MinGW [SSL] [LZO] built on Oct  1 2006
Fri Aug 06 11:29:57 2010 WARNING: No server certificate verification method has been enabled.  See http://openvpn.net/
Fri Aug 06 11:29:57 2010 LZO compression initialized
Fri Aug 06 11:29:57 2010 Control Channel MTU parms [ L:1542 D:138 EF:38 EB:0 ET:0 EL:0 ]
Fri Aug 06 11:29:57 2010 Data Channel MTU parms [ L:1542 D:1450 EF:42 EB:135 ET:0 EL:0 AF:3/1 ]
Fri Aug 06 11:29:57 2010 Local Options hash (VER=V4): '41690919'
Fri Aug 06 11:29:57 2010 Expected Remote Options hash (VER=V4): '530fdded'
Fri Aug 06 11:29:57 2010 UDPv4 link local: [undef]
Fri Aug 06 11:29:57 2010 UDPv4 link remote: 192.168.250.152:1194
Fri Aug 06 11:29:57 2010 TLS: Initial packet from 192.168.250.152:1194, sid=dd3359de 265f251d
Fri Aug 06 11:30:01 2010 VERIFY OK: depth=1, /C=US/ST=CA/L=SanFrancisco/O=Fort-Funston/CN=OpenVPN-CA/emailAddress=me@m
Fri Aug 06 11:30:01 2010 VERIFY OK: depth=0, /C=US/ST=CA/L=SanFrancisco/O=Fort-Funston/CN=server/emailAddress=me@myhos
Fri Aug 06 11:30:02 2010 Data Channel Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Fri Aug 06 11:30:02 2010 Data Channel Encrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Fri Aug 06 11:30:02 2010 Data Channel Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Fri Aug 06 11:30:02 2010 Data Channel Decrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Fri Aug 06 11:30:02 2010 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 1024 bit RSA
Fri Aug 06 11:30:02 2010 [server] Peer Connection Initiated with 192.168.250.152:1194
Fri Aug 06 11:30:04 2010 SENT CONTROL [server]: 'PUSH_REQUEST' (status=1)
Fri Aug 06 11:30:04 2010 PUSH: Received control message: 'PUSH_REPLY,route 10.100.10.1,topology net30,ping 10,ping-res
Fri Aug 06 11:30:04 2010 Options error: Unrecognized option or missing parameter(s) in [PUSH-OPTIONS]:2: topology (2.0
Fri Aug 06 11:30:04 2010 OPTIONS IMPORT: timers and/or timeouts modified
Fri Aug 06 11:30:04 2010 OPTIONS IMPORT: --ifconfig/up options modified
Fri Aug 06 11:30:04 2010 OPTIONS IMPORT: route options modified
Fri Aug 06 11:30:04 2010 TAP-WIN32 device [Local Area Connection 3] opened: \\.\Global\{12EF532A-3135-4F37-B689-720FEC
Fri Aug 06 11:30:04 2010 TAP-Win32 Driver Version 8.4
Fri Aug 06 11:30:04 2010 TAP-Win32 MTU=1500
Fri Aug 06 11:30:04 2010 Notified TAP-Win32 driver to set a DHCP IP/netmask of 10.100.10.6/255.255.255.252 on interfac
Fri Aug 06 11:30:04 2010 Successful ARP Flush on interface [5] {12EF532A-3135-4F37-B689-720FE0B1F713}
Fri Aug 06 11:30:04 2010 TEST ROUTES: 0/0 succeeded len=1 ret=0 a=0 u/d=down
Fri Aug 06 11:30:04 2010 Route: Waiting for TUN/TAP interface to come up...
Fri Aug 06 11:30:05 2010 TEST ROUTES: 0/0 succeeded len=1 ret=0 a=0 u/d=down
Fri Aug 06 11:30:05 2010 Route: Waiting for TUN/TAP interface to come up...
Fri Aug 06 11:30:06 2010 TEST ROUTES: 1/1 succeeded len=1 ret=1 a=0 u/d=up
```

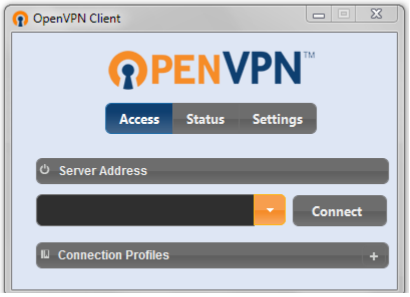
- Once established, the OpenVPN icon will display a message notifying of the successful connection and assigned IP. This information, as well as the time the connection was established, is available anytime by scrolling over the OpenVPN icon.



client is now connected.

Assigned IP: 10.100.10.6

***Note**: An alternate OpenVPN Windows client can be downloaded from **http://www.openvpn.net/index.php/openvpn-client/downloads.html**. Refer to **http://www.openvpn.net/index.php/openvpn-client/howto-openvpn-client.html** for help*



4.11 PPTP VPN

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.

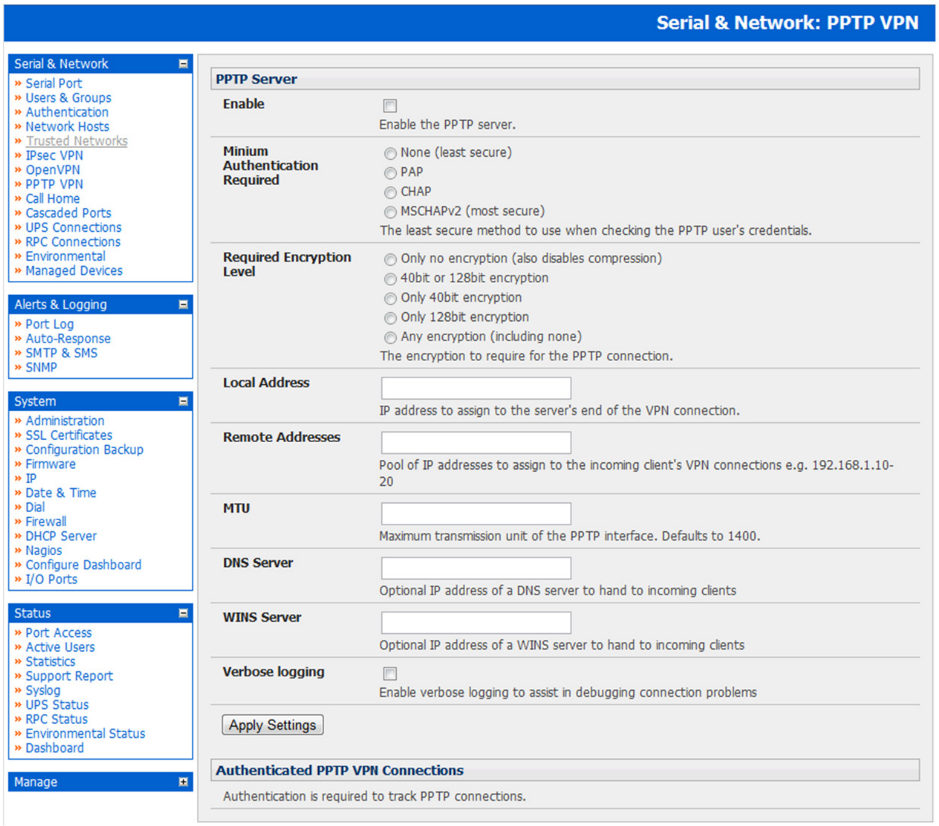
The strength of PPTP is its ease of configuration and integration into existing Microsoft infrastructure. It is generally used for connecting single remote Windows clients. If you take your portable computer on a business trip, you can dial a local number to connect to your Internet service provider (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 appliance and local network:

1. Enable and configure the PPTP VPN server on your appliance
2. Set up VPN user accounts on your appliance and enable the appropriate authentication
3. 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 XP/ NT/ 2000/ 7 and Vista
4. Connect to the remote VPN

4.11.1 Enable the PPTP VPN server

- Select **PPTP VPN** on the **Serial & Networks** menu



PPTP Server

Enable □
Enable the PPTP server.

Minium Authentication Required
○ None (least secure)
○ PAP
○ CHAP
○ MSCHAPv2 (most secure)
The least secure method to use when checking the PPTP user's credentials.

Required Encryption Level
○ Only no encryption (also disables compression)
○ 40bit or 128bit encryption
○ Only 40bit encryption
○ Only 128bit encryption
○ Any encryption (including none)
The encryption to require for the PPTP connection.

Local Address
IP address to assign to the server's end of the VPN connection.

Remote Addresses
Pool of IP addresses to assign to the incoming client's VPN connections e.g. 192.168.1.10-20

MTU
Maximum transmission unit of the PPTP interface. Defaults to 1400.

DNS Server
Optional IP address of a DNS server to hand to incoming clients

WINS Server
Optional IP address of a WINS server to hand to incoming clients

Verbose logging □
Enable verbose logging to assist in debugging connection problems

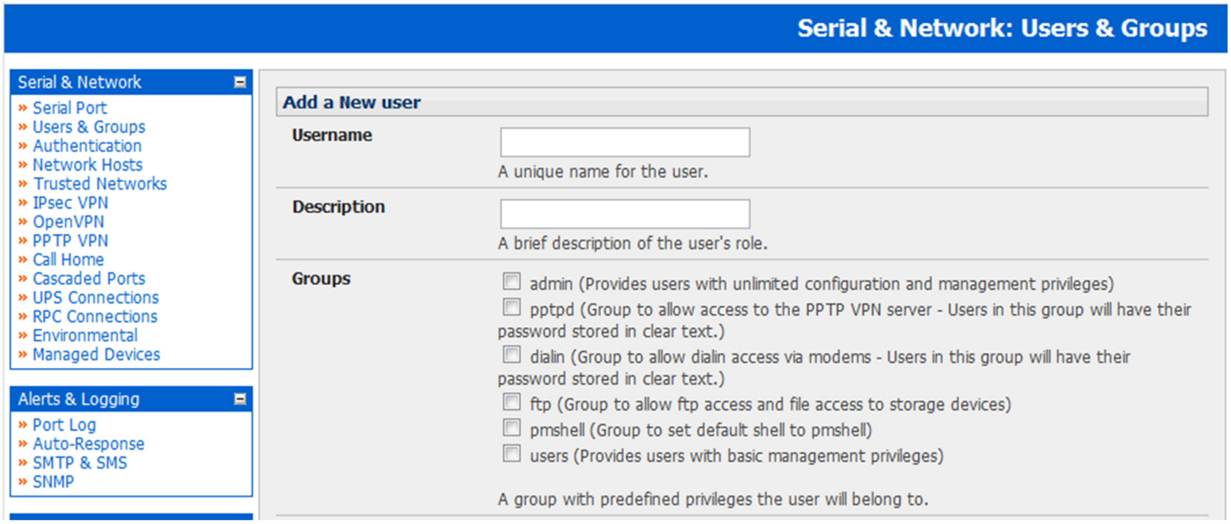
Authenticated PPTP VPN Connections

Authentication is required to track PPTP connections.

- Select the **Enable** check box to enable the PPTP Server
- Select the **Minimum Authentication Required**. Access is denied to remote users attempting to connect using an authentication scheme weaker than the selected scheme. The schemes are described below, from strongest to weakest.
    - o **Encrypted Authentication (MS-CHAP v2)**: The strongest type of authentication to use; this is the recommended option
    - o **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
    - o **Unencrypted Authentication (PAP)**: This is plain text password authentication. When using this type of authentication, the client password is transmitted unencrypted.
    - o **None**
- Select the **Required Encryption Level**. Access is denied to remote users attempting to connect not using this encryption level. Strong 40 bit or 128 bit encryption is recommended
- In **Local Address**, enter 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 (e.g. 192.168.1.10-20). This must be a free IP address (or a range of free IP addresses), from the network (typically the LAN) that remote users are assigned while connected to the 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 Settings**

4.11.2 Add a PPTP user

- Select **Users & Groups** on the **Serial & Networks** menu and complete the fields as covered in section 4.2.
- 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.
- Keep note of the username and password for when you need to connect to the VPN connection
- Click **Apply**



Add a New user

Username

A unique name for the user.

Description

A brief description of the user's role.

Groups

□ 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 dialin access via modems - 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)

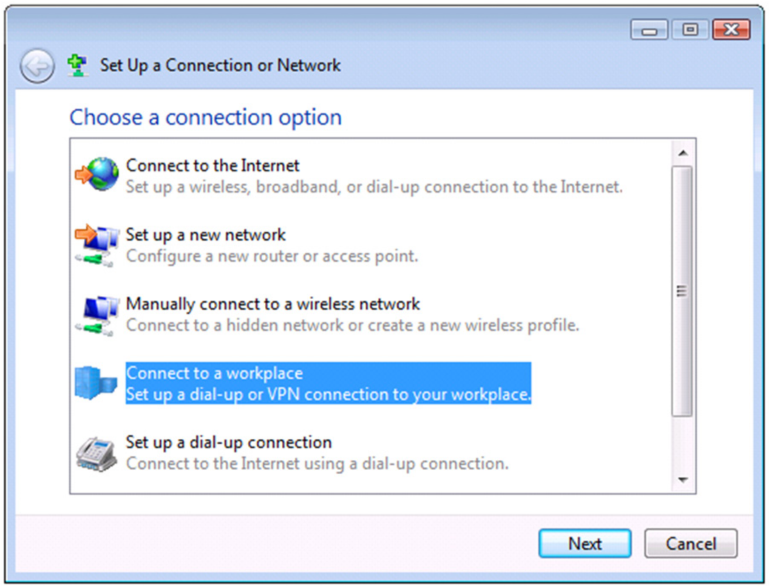
A group with predefined privileges the user will belong to.

4.11.3 Set up 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 appliance.

***Note**: This procedure sets up a PPTP client in the Windows 7 Professional operating system. The steps may vary slightly depending on your network access or if you are using an alternate version of Windows. More detailed instructions are available from the Microsoft web site.*

- Login to your Windows client with administrator privileges
- From the **Network & Sharing Center** on the **Control Panel** select **Network Connections** and create a new connection



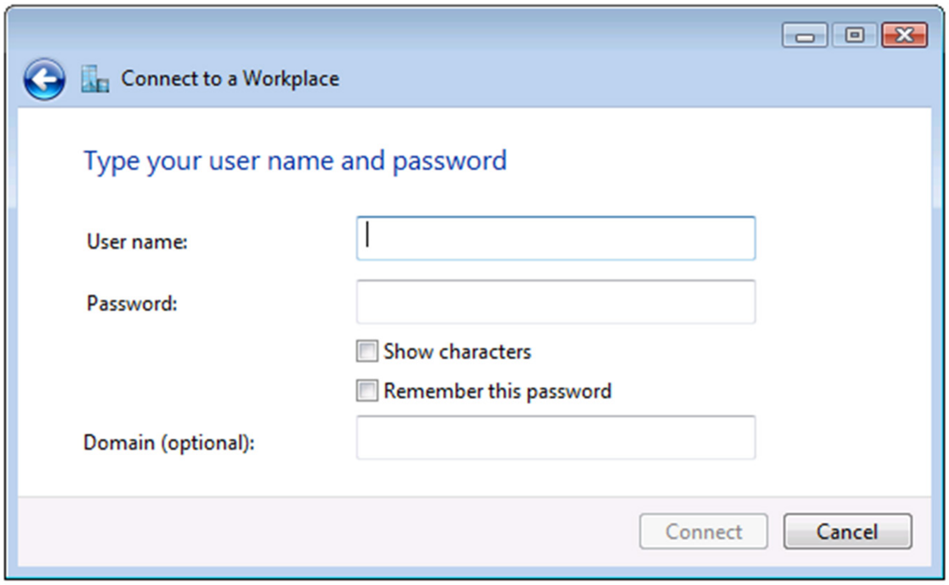
Set Up a Connection or Network

Choose a connection option

- Connect to the Internet
  Set up a wireless, broadband, or dial-up connection to the Internet.
- Set up a new network
  Configure a new router or access point.
- Manually connect to a wireless network
  Connect to a hidden network or create a new wireless profile.
- Connect to a workplace
  Set up a dial-up or VPN connection to your workplace.
- Set up a dial-up connection
  Connect to the Internet using a dial-up connection.

- Select **Use My Internet Connection (VPN)** and enter the IP Address of the 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 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.*



Connect to a Workplace

Type your user name and password

User name:

Password:

□ Show characters

□ Remember this password

Domain (optional):

Chapter 5: Firewall, Failover and Out-of-Band

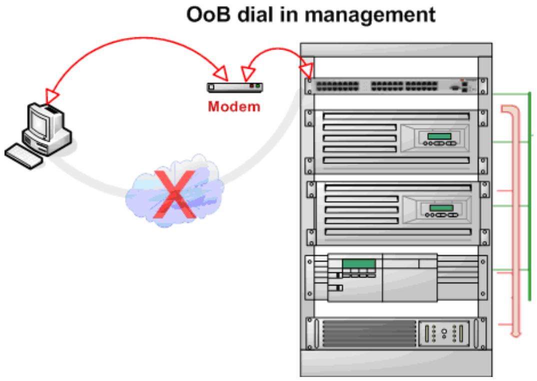
The Console Server has a number of failover and out-of-band access capabilities to ensure availability in the event there are difficulties in accessing the Console Server through the principal network path. This chapter covers:

- Out-of-band (OoB) access from a remote location using dial-up modem
- Out-dial failover
- OoB access using an alternate broadband link
- Broadband failover

The Console Server can also provide basic routed firewall facilities with NAT (Network Address Translation), packet filtering and port forwarding support on all network interfaces.

5.1 OoB Dial-In Access

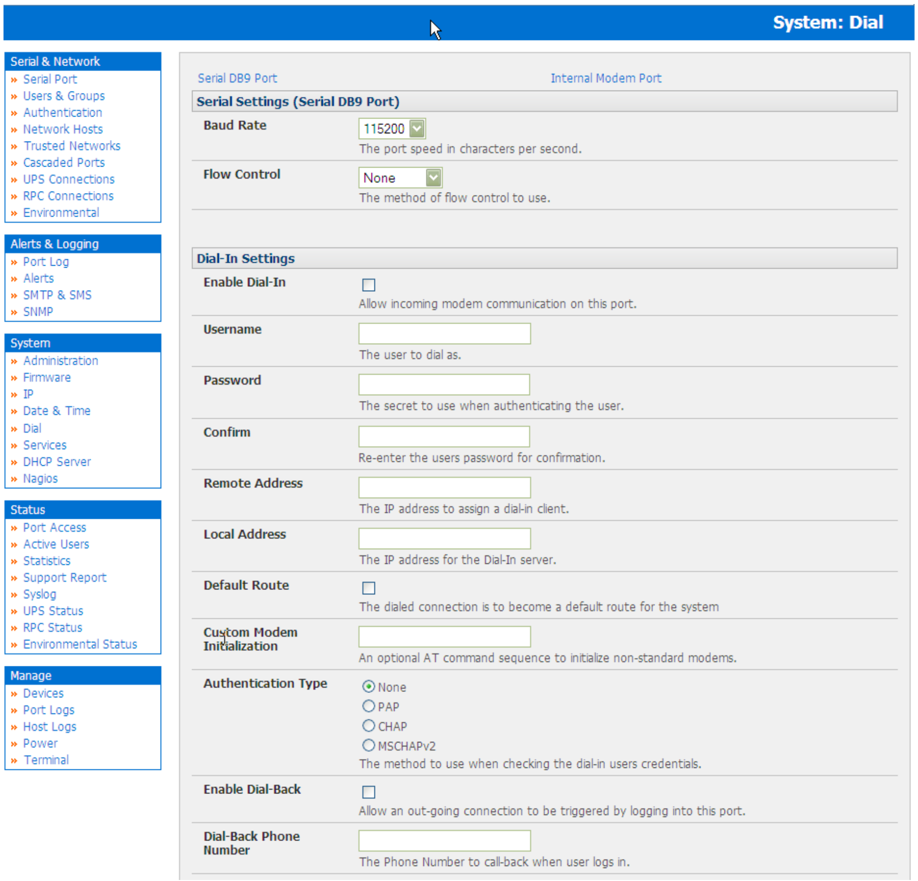
To enable OoB dial-in access, first set up the Console Server configuration for dial-in PPP access. Once the Console Server is so configured, it will wait for an incoming connection from a dial-in at a remote site.
Then remote Administrator's must be configured to dial-in and must establish a network connection to the Console Server.



***Note**: The B094-008-2E-M-F, B096-048/032/016 and BO095-003-M Console Servers have an internal modem for dial-up OoB access. The B092-016 Console Server needs an external modem to be attached via a serial cable to its DB9 port. With the B095-004 Console Server the four serial ports are by default all configured as RJ serial Console Server ports. However Port 1 can be configured to be the Local Console/Modem port for an external modem to be attached.*

5.1.1 Configure dial-in PPP

To enable dial-in PPP access on the Console Server modem port/ internal modem:



Serial Settings (Serial DB9 Port)

Baud Rate 115200
The port speed in characters per second.

Flow Control None
The method of flow control to use.

Dial-In Settings

Enable Dial-In □
Allow incoming modem communication on this port.

Username
The user to dial as.

Password
The secret to use when authenticating the user.

Confirm
Re-enter the users password for confirmation.

Remote Address
The IP address to assign a dial-in client.

Local Address
The IP address for the Dial-In server.

Default Route □
The dialed connection is to become a default route for the system

Custom Modem Initialization
An optional AT command sequence to initialize non-standard modems.

Authentication Type
● None
○ PAP
○ CHAP
○ MSCHAPv2
The method to use when checking the dial-in users credentials.

Enable Dial-Back □
Allow an out-going connection to be triggered by logging into this port.

Dial-Back Phone Number
The Phone Number to call-back when user logs in.

- Select the **System: Dial** menu option and the port to be configured (**Serial DB9 Port** or **Internal Modem Port**)

***Note**: The Console Server's console/modem serial port is set by default to 115200 baud, No parity, 8 data bits and 1 stop bit, with software (Xon-Xoff) flow control enabled. You can modify the baud rate and flow control using the Management Console. You can further configure the console/modem port settings by editing **/etc/mgetty.config** files as described in **Chapter 14**.*

- Select the **Baud Rate** and **Flow Control** that will communicate with the modem
- Check the **Enable Dial-In Access** box
- Enter the **User name** and **Password** to be used for the dial-in PPP link
- 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. However, it and the Local IP Address must both be in the same network range (e.g. 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 both must be in the same network range as the Remote IP Address
- The **Default Route** option enables the dialed PPP connection to become the default route for the Console Server
- The **Custom Modem Initialization** option allows a custom AT string modem initialization string to be entered (e.g. AT&C1&D3&K3)
- Then select the **Authentication Type** to be applied to the dial-in connection. The Console Server uses authentication to challenge Administrators who dial-in to the Console Server. (For dial-in access, the username and password received from the dial-in client are verified against the local authentication database stored on the Console Server). The Administrator must also have their client computer configured to use the selected authentication scheme. Select **PAP CHAP MSCHAPv2** or **None** and click **Apply**

None With this selection, no username or password authentication is required for dial-in access. This is not recommended.

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.

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.

MSCHAPv2 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

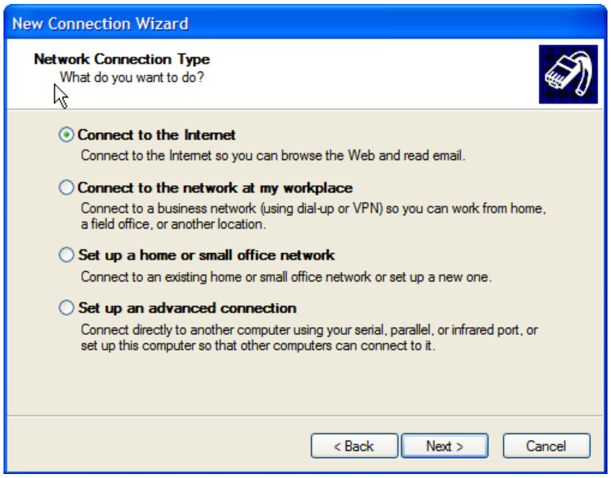
- Console Servers all support dial-back for additional security. This is configured per-user in **Serial & Network: Users & Groups → Edit**. Check the **Enable Dial-Back** box and enter the phone number to be called to re-establish an OoB link once a dial-in connection has been logged

5.1.2 Using SDT Connector client for dial-in

Administrators can use their SDT Connector client to set up secure OoB dial-in access to all their remote Console Servers. With a point and click you can initiate a dial-up connection. Refer to Chapter 6.5.

5.1.3 Set up Windows XP/ 2003/Vista/7 client for dial-in

- Open **Network Connections** in Control Panel and click the **New Connection Wizard**



New Connection Wizard

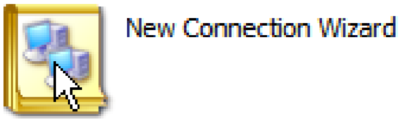
Network Connection Type
What do you want to do?

● Connect to the Internet
Connect to the Internet so you can browse the Web and read email.

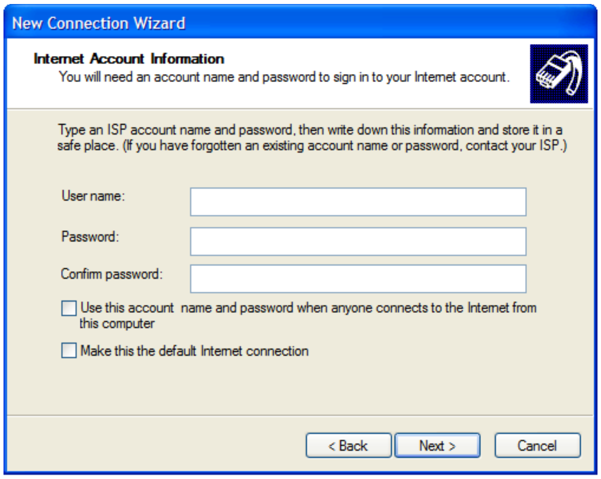
○ Connect to the network at my workplace
Connect to a business network (using dial-up or VPN) so you can work from home, a field office, or another location.

○ Set up a home or small office network
Connect to an existing home or small office network or set up a new one.

○ Set up an advanced connection
Connect directly to another computer using your serial, parallel, or infrared port, or set up this computer so that other computers can connect to it.



- Select **Connect to the Internet** and click **Next**
- On the **Getting Ready** screen select **Set Up My Connection Manually** and click **Next**
- On the **Internet Connection** screen select **Connect Using a Dial-Up Modem** and click **Next**
- Enter a **Connection Name** (any name you choose) and the dial-up **Phone Number** that will connect thru to the Console Server modem



New Connection Wizard

Internet Account Information

You will need an account name and password to sign in to your Internet account.

Type an ISP account name and password, then write down this information and store it in a safe place. (If you have forgotten an existing account name or password, contact your ISP.)

User name:

Password:

Confirm password:

□ Use this account name and password when anyone connects to the Internet from this computer

□ Make this the default Internet connection

- Enter the PPP **User Name** and **Password** for have set up for the Console Server

5.1.4 Set up earlier Windows clients for dial-in

- 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 the **Start** button and selecting **Settings**. Then click **Network and Dial-up Connections** and click **Make New Connection**
- Similarly, for Windows 98, you double-click **My Computer** on the Desktop, then open **Dial-Up Networking** and double click **Make New Connection** and proceed as above

5.1.5 Set up Linux clients for dial-in

The online tutorial *http://www.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/ifdown* to start and stop a PPP connection
- Using the Gnome control panel configuration tool -
- WVDIAL and the Redhat "Dialup configuration tool"
- GUI dial program X-isp. Download/Installation/Configuration

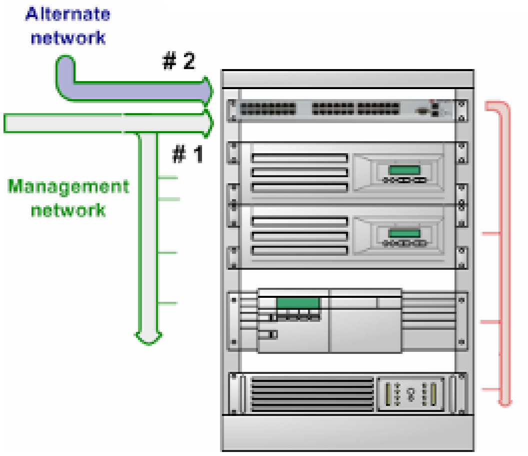
***Note**: For all PPP clients:*

- *Set the PPP link up with TCP/IP as the only protocol enabled*
- *Specify that the Server will assign IP address and do DNS*
- *Do not set up the Console Server PPP link as the default for Internet connection*

5.2 OoB Broadband Access

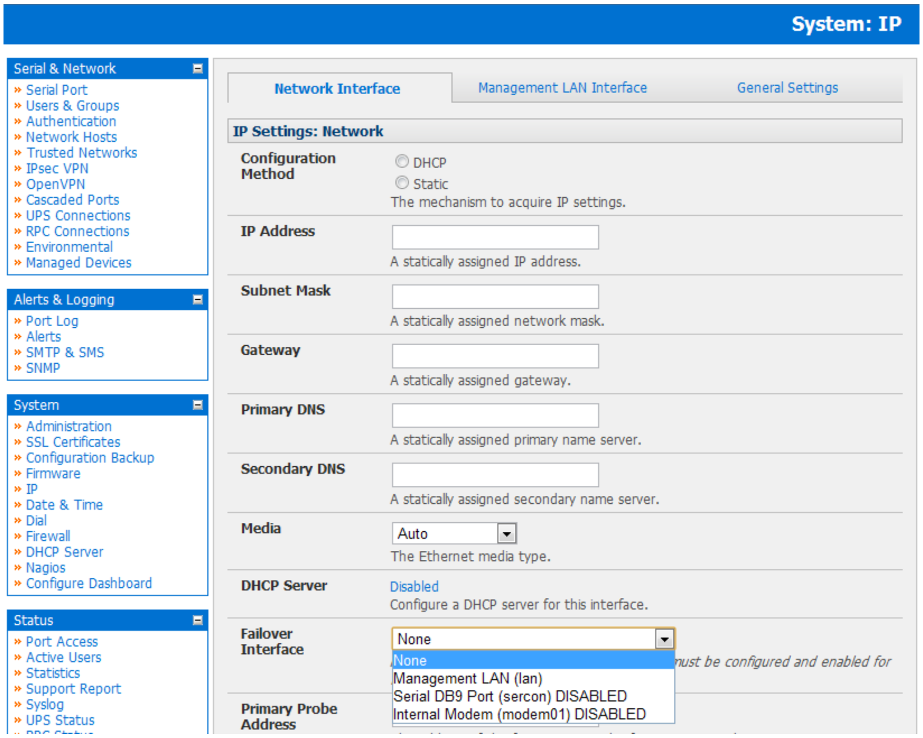
The B096-048/032/016 Console Server Management Switch has a second Ethernet network port that can be configured for alternate and OoB (out-of-band) broadband access. With two active broadband access paths to the Console Server, in the event you are unable to access through the primary management network, you may still have access through the alternate broadband path (e.g. a T1 link).

- On the **System: IP** menu, select **Management LAN Interface** and configure the **IP Address, Subnet Mask, Gateway** and **DNS** with the access settings that relate to the alternate link
- Ensure that when configuring the principal **Network Interface** connection, you set the **Failover Interface** to *None*



5.3 Broadband Ethernet Failover

The second Ethernet port on the B096-048/032/016 Console Server Management Switch can also be configured for failover to ensure transparent high availability.



IP Settings: Network

Configuration Method ○ DHCP ○ Static
The mechanism to acquire IP settings.

IP Address
A statically assigned IP address.

Subnet Mask
A statically assigned network mask.

Gateway
A statically assigned gateway.

Primary DNS
A statically assigned primary name server.

Secondary DNS
A statically assigned secondary name server.

Media: Auto
The Ethernet media type.

DHCP Server: Disabled
Configure a DHCP server for this interface.

Failover Interface: None
None
Management LAN (lan)
Serial DB9 Port (sercon) DISABLED
Internal Modem (modem01) DISABLED
nust be configured and enabled for

Primary Probe Address

- When configuring the principal network connection on the **System: IP Network Interface** menu, select **Management LAN** (eth1) as the **Failover Interface** to be used when a fault has been detected with main Network Interface (eth0)

- Specify the **Probe Addresses** of two sites (the **Primary** and **Secondary**) that the B096-048/032/016 is to *ping* to determine if Network (eth0) is still operational
- Then configure **Management LAN Interface (eth1)** with the same IP setting that you used for the main **Network Interface (eth0)** to ensure transparent redundancy

In this mode, Network 2 (eth1) is available as the transparent back-up port to Network 1 (eth0) for accessing the management network. Network 2 will automatically and transparently take over the work of Network 1, in the event Network 1 becomes unavailable for any reason.

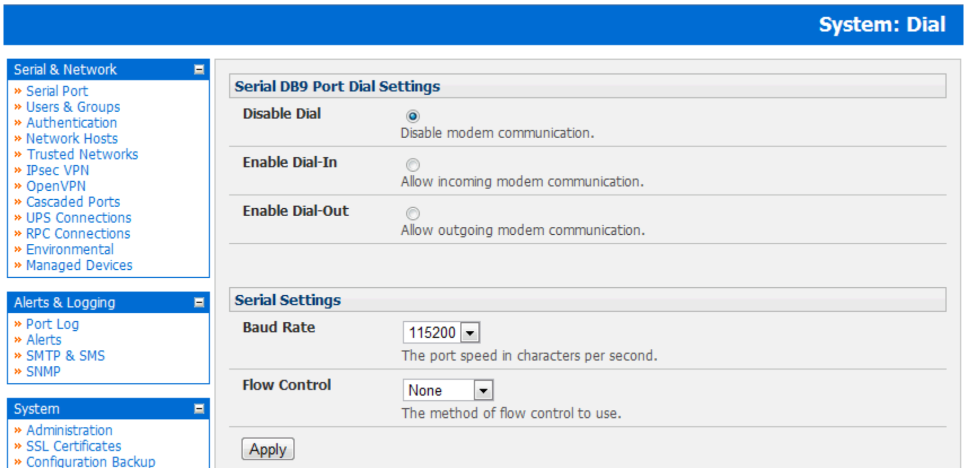
By default, the Console Server supports automatic failure-recovery back to the original state prior to failover. The Console Server continually pings probe addresses whilst in original and failover states. The original state will automatically be set as a priority and re-established following three successful pings of the probe addresses during failover. The failover state will be removed once the original state has been re-established.

5.4 Dial-Out Access

The internal or externally attached modem on the Console Servers can be set up either

o in *Failover mode*, where a dial-out connection is only established in event of a *ping* failure, or

o with the dial-out connection always on



Serial DB9 Port Dial Settings

Disable Dial ● Disable modem communication.

Enable Dial-In ○ Allow incoming modem communication.

Enable Dial-Out ○ Allow outgoing modem communication.

Serial Settings

Baud Rate 115200 The port speed in characters per second.

Flow Control None The method of flow control to use.

In both of the above cases, in the event of a disruption in the dial-out connection, the Console Server will endeavor to re-establish the connection.

5.4.1 Always-on dial-out

The Console Server modem can be configured for out-dial to be always on, with a permanent external dial-up ppp connection.

- Select the **System: Dial** menu option and check **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** field enter the access details for the remote PPP server to be called

Override DNS 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 box. Enter the IP of the DNS servers into the spaces provided.

5.4.2 Dial-Out Failover

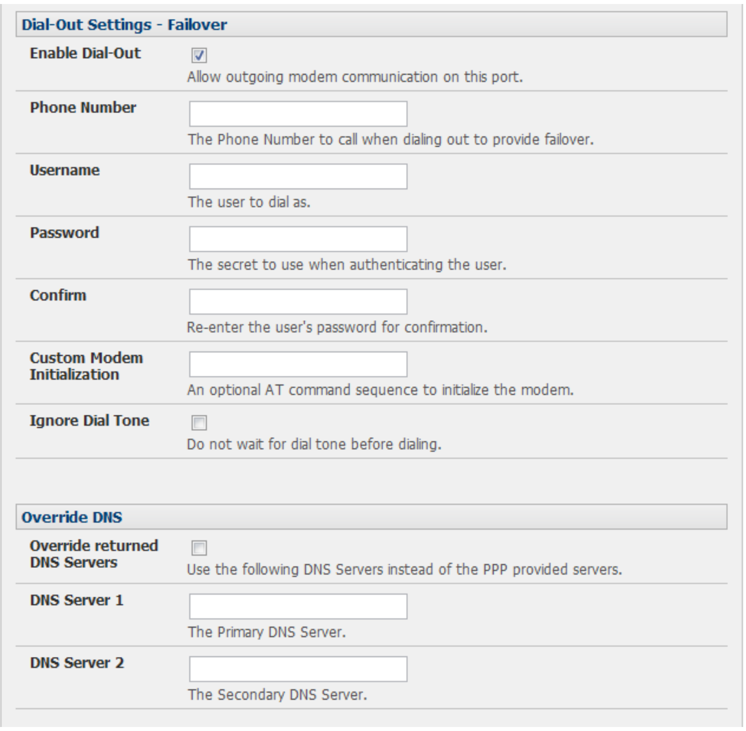
The Console Servers can also be configured for dial-out failover— so a dial-out PPP connection is automatically set up in the event of a disruption in the principal management network:

- When configuring the principal network connection in **System: IP**, specify **Internal Modem** (or the **Dial Serial DB9** if using an external modem on the Console port) as the **Failover Interface** to be used when a fault has been detected with Network1 (eth0)
- Specify the **Probe Addresses** of two sites (the **Primary** and **Secondary**) that the Console Server is to *ping* to determine if Network1 is still operational
- Select the **System: Dial** menu option and the port to be configured (**Serial DB9 Port** or **Internal Modem Port**)
- Select the **Baud Rate** and **Flow Control** that will communicate with the modem

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 13.*

- Check the **Enable Dial-Out Access** box and enter the access details for the remote PPP server to be called

Note*: Both SSH and HTTPS access is enabled for dial-out failover, do the administrator can SSH (or HTTPS) connect to the console server (and its Managed Devices) and fix the problem*



Dial-Out Settings - Failover

Enable Dial-Out	☑ Allow outgoing modem communication on this port.
Phone Number	The Phone Number to call when dialing out to provide failover.
Username	The user to dial as.
Password	The secret to use when authenticating the user.
Confirm	Re-enter the user's password for confirmation.
Custom Modem Initialization	An optional AT command sequence to initialize the modem.
Ignore Dial Tone	□ Do not wait for dial tone before dialing.

Override DNS

Override returned DNS Servers	□ Use the following DNS Servers instead of the PPP provided servers.
DNS Server 1	The Primary DNS Server.
DNS Server 2	The Secondary DNS Server.

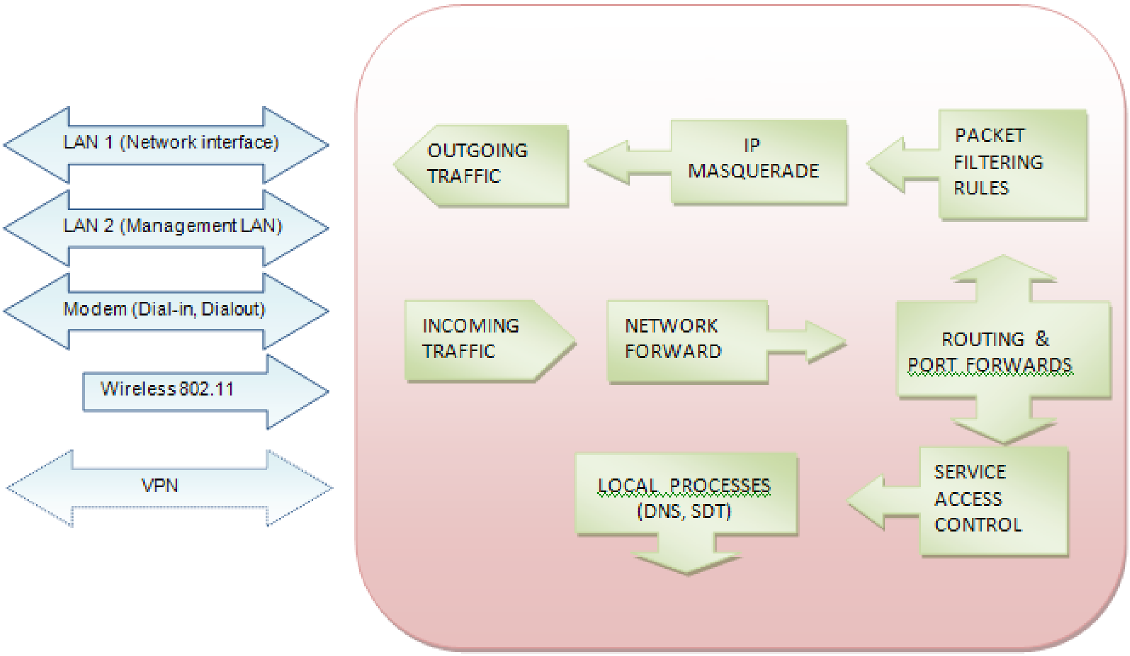
Override DNS is available for PPP Devices such as modems. Override DNS allows the use of alternate DNS servers from those provided by your ISP. For example, an alternative DNS may be required for OpenDNS used for content filtering.

- To enable **Override DNS**, check the **Override returned DNS Servers** box. Enter the IP of the DNS servers into the spaces provided

Note*: By default, the Console Server supports automatic failure-recovery back to the original state prior to failover. The Console Server continually pings probe addresses whilst 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.*

5.5 Firewall & Forwarding

Console Servers provide basic firewalled routing, NAT (Network Address Translation), packet filtering and port forwarding support on all network interfaces.



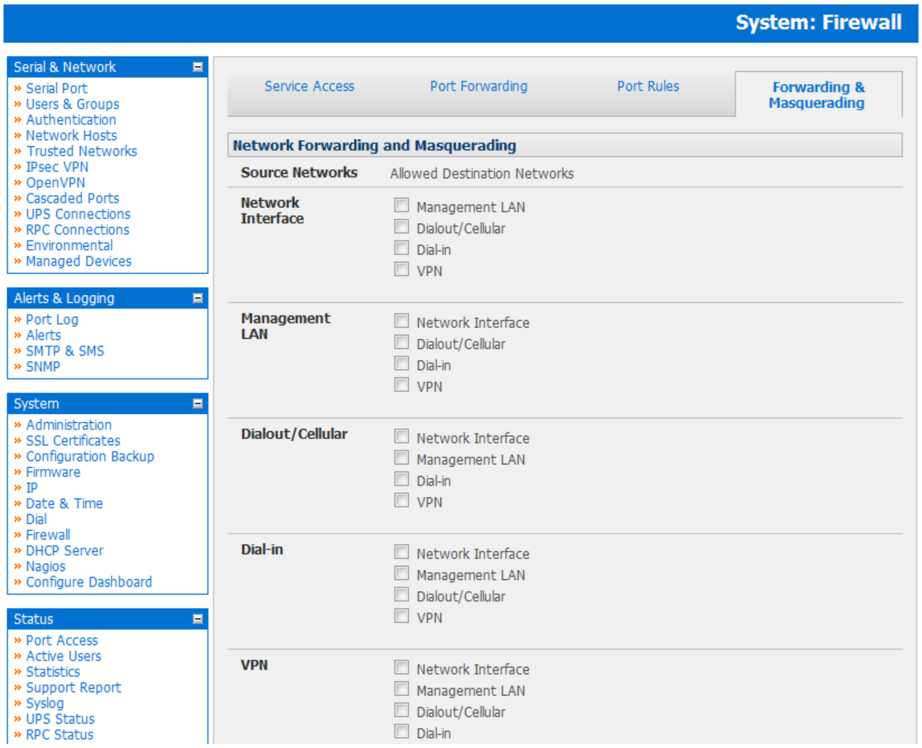
5.5.1 Configuring network forwarding and IP masquerading

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*.

Note*: Network* ***forwarding*** *allows the network packets on one network interface (i.e. LAN1/ eth0) to be forwarded to another network interface (i.e. LAN2/eth1 or dial-out/cellular) so that locally networked devices can connect to IP 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.*

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/external network:

- Navigate to the **System: Firewall** page, and then click on the **Forwarding &Masquerading** tab



Network Forwarding and Masquerading

Source Networks	Allowed Destination Networks
Network Interface	□ Management LAN □ Dialout/Cellular □ Dial-in □ VPN
Management LAN	□ Network Interface □ Dialout/Cellular □ Dial-in □ VPN
Dialout/Cellular	□ Network Interface □ Management LAN □ Dial-in □ VPN
Dial-in	□ Network Interface □ Management LAN □ Dialout/Cellular □ VPN
VPN	□ Network Interface □ Management LAN □ Dialout/Cellular □ Dial-in

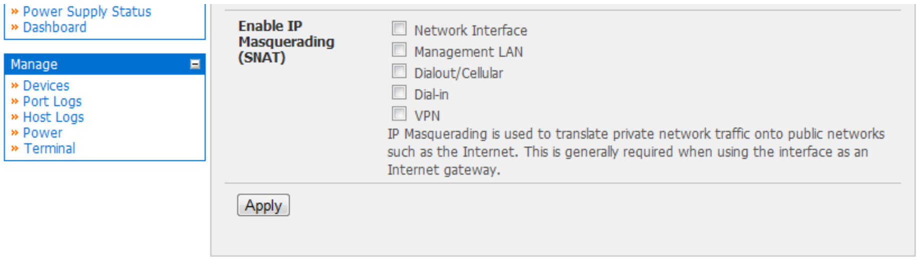
- Find the **Source Network** to be routed, and then tick the relevant **Destination Network** to enable Forwarding

For example to configure a dual Ethernet device such as a B096-048, B096-032 or B096-016 Console Server Management Switch:

- The **Source Network** would the **Network Interface** and the **Destination Network** would be **Management LAN**

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 will translate 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.



Enable IP Masquerading (SNAT)	□ Network Interface □ Management LAN □ Dialout/Cellular □ Dial-in □ VPN IP Masquerading is used to translate private network traffic onto public networks such as the Internet. This is generally required when using the interface as an Internet gateway.

By default IP Masquerading is disabled for all networks. To enable masquerading:

- Select **Forwarding & Masquerading** panel on the **System: Firewall** menu
- Check **Enable IP Masquerading (SNAT)** on the network interfaces where masquerading is be enabled

Generally this masquerading would be applied to any interface that is connecting with a public network such as the Internet.

5.5.2 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 using DHCP

Manual Configuration:

Manually set a static gateway address (being the address of the Console Server) and set the DNS server address to be the same as used on the external network i.e. if the Console Server is acting as an internet gateway or a cellular router, then use the ISP provided DNS server address.

DHCP Configuration:

- Navigate to the System:IP page
- Click the tab of the interface connected to the internal network. To use DHCP, a static address must be set; check that the static IP and subnet mask fields are set.
- Click on the Disabled link next to DHCP Server which will bring up the System: DHCP Server page
- Check Enable DHCP Server
- To configure the DHCP server, tick the Use interface address as gateway check box
- Set the DNS server address(es), lease times, allocation pools and pre-assigned IP addresses; as detailed previously in Chapter 3.6.2

Once applied, devices on the internal network will be able to access resources on the external network.

5.5.3 Port/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, and have the Console Server redirect the data to a specified internal address and port range.

To setup a port forward:

- Navigate to the **System: Firewall** page, and click on the **Port Forwarding** tab
- Click **Add New Port Forward**
- Fill in the following fields:

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"

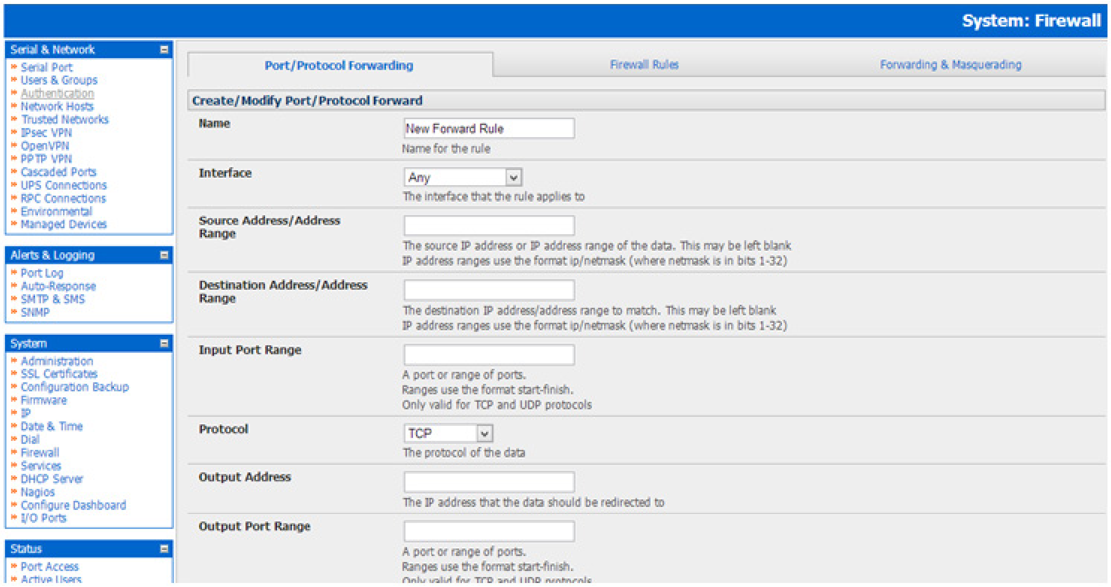
Source Address/
Address Range: This allows the user to restrict access to a port forward to a specific source IP address or IP address range of the data. This may be left blank. IP address ranges use the format ip/netmask (where netmask is in bits 1-32).

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 or UDP, TCP and UDP, ICMP or ESP* or *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 ports that the packets will be redirected to on the Output Address.



Create/Modify Port/Protocol Forward

Name	New Forward Rule Name for the rule
Interface	Any The interface that the rule applies to
Source Address/Address Range	The 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	A port or range of ports. Ranges use the format start-finish. Only valid for TCP and UDP protocols
Protocol	TCP The protocol of the data
Output Address	The IP address that the data should be redirected to
Output Port Range	A port or range of ports. 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, the following settings would be used:

Input Interface: Any

Input Port Range: 8443

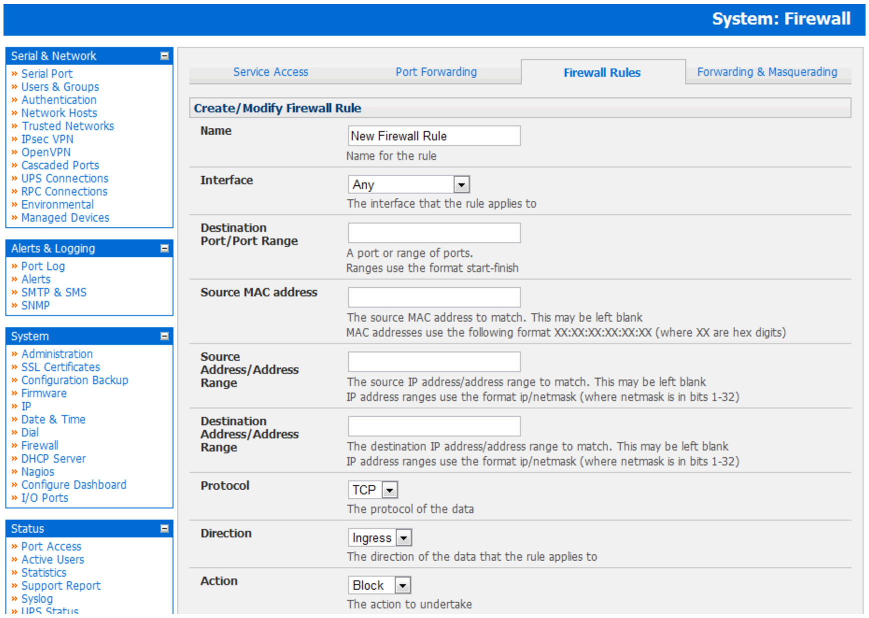
Protocol: TCP

Output Address: 192.168.10.2

Output Port Range: 443

5.5.4 Firewall Rules

Firewall rules can be used to block or allow traffic through an interface based on port number, direction (ingress or egress) and protocol. This can be used to allow custom on box services, or block traffic based on policy.



Create/Modify Firewall Rule

Name	New Firewall Rule Name for the rule
Interface	Any The interface that the rule applies to
Destination Port/Port Range	A port or range of ports. Ranges use the format start-finish
Source MAC address	The source MAC address to match. This may be left blank MAC addresses use the following format XX:XX:XX:XX:XX:XX (where XX are hex digits)
Source Address/Address Range	The source 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)
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)
Protocol	TCP The protocol of the data
Direction	Ingress The direction of the data that the rule applies to
Action	Block The action to undertake

To setup a firewall rule:

- Navigate to the **System: Firewall** page, and click on the **Firewall Rules** tab
- Click **Add New Firewall Rule**
- Fill in the following fields:

Name: Name the firewall rule. This name should describe the policy the port rule is being used to implement (e.g. *block ftp*)

Interface: Select the interface that the firewall rule will be applied to (i.e. *Any, Dialout/Cellular, VPN, Network Interface, Dial-in* etc)

Port Range: Specify the port or range of ports (e.g. 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

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 or UDP

Direction: Select the traffic direction that the firewall rule will apply to (*Ingress* = incoming or *Egress*)

Action: Select the action (*Accept* or *Block*) that will be applied to the packets detected that match the Interface+ Port Range + Source/destination Address Range + Protocol+ Direction

For example, to block SSH traffic from leaving Dialout Interface, the following settings can be used:

Interface: Dialout

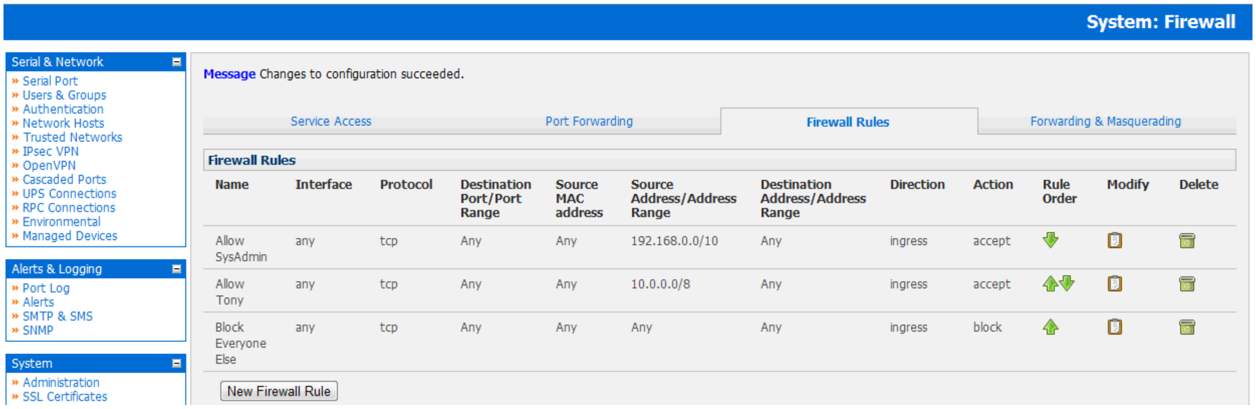
Port Range: 22

Protocol: TCP

Direction: Egress

Action: Block

The 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*):



System: Firewall

Message Changes to configuration succeeded.

Firewall Rules

Name	Interface	Protocol	Destination Port/Port Range	Source MAC address	Source Address/Address Range	Destination Address/Address Range	Direction	Action	Rule Order	Modify	Delete
Allow SysAdmin	any	tcp	Any	Any	192.168.0.0/10	Any	ingress	accept	↓		
Allow Tony	any	tcp	Any	Any	10.0.0.0/8	Any	ingress	accept	↑↓		
Block Everyone Else	any	tcp	Any	Any	Any	Any	ingress	block	↑		

	To allow all incoming traffic on all interfaces from the SysAdmin:	To allow all incoming traffic from Tony:	To block all incoming traffic from the Network Interface:
Interface	Any	Any	Network Interface
Port Range	Any	Any	Any
Source MAC	Any	Any	Any
Source IP	IP address of SysAdmin	IP address of Tony	Any
Destination IP	Any	Any	Any
Protocol	TCP	TCP	TCP
Direction	Ingress	Ingress	Ingress
Action	Accept	Accept	Block

However, if the **Rule Order** above was to be changed so the "*Block Everyone Else*" rule was second on the list, then the traffic coming in over the *Network Interface* from *Tony* would be blocked.

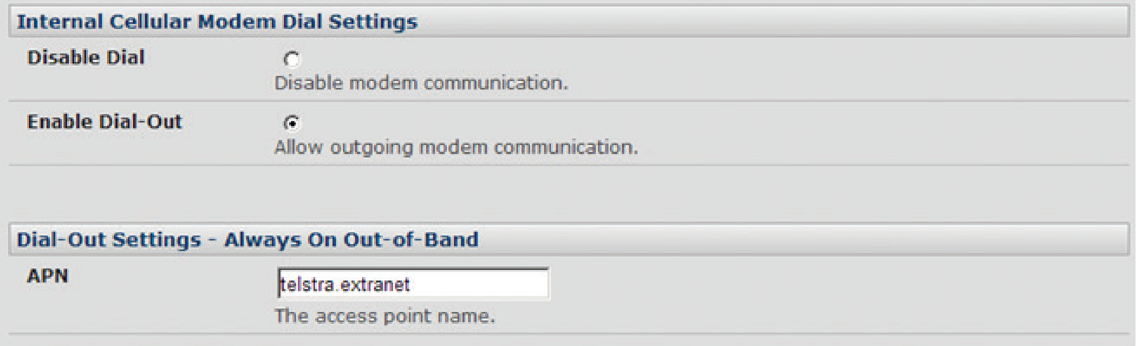
5.6 Internal Cellular Modem Connection

5.6.1 Connecting to a *4G LTE* carrier network

The B094-008-2E-V has an internal cellular modem that will connect to Verizon's 4G LTE network (USA).

- Before powering on the B094-008-2E-V, you must first install the SIM card provided by your cellular carrier and attach the external aerial antenna.

- Select **Internal Cellular Modem** panel on the **System: Dial** menu.
- Check **Enable Dial-Out Settings**.



Internal Cellular Modem Dial Settings

Disable Dial ○
Disable modem communication.

Enable Dial-Out ●
Allow outgoing modem communication.

Dial-Out Settings - Always On Out-of-Band

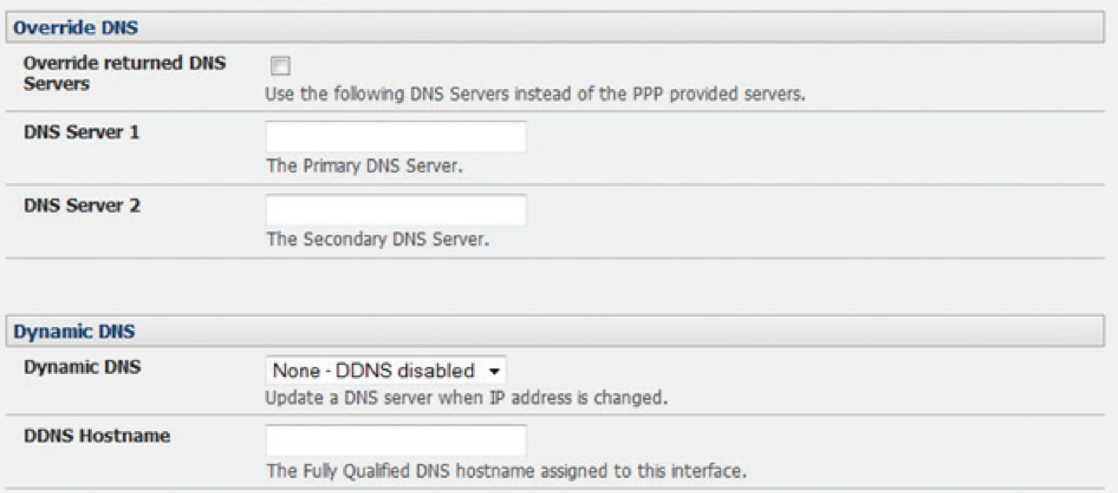
APN: telstra.extranet
The access point name.

Note: *Your 4G LTE carrier may have provided you with details for configuring the connection, including APN (Access Point Name), PIN code (optional PIN code that may be required to unlock the SIM card), Username/Password, etc. In most cases, you will only need to enter your cellular provider's APN, leaving the other fields blank.*

- Enter the carrier's **APN**.
- If the SIM card is configured with a PIN code, you will be required to enter a PIN code to unlock the card.

You may also need to set Override DNS to use alternate DNS servers from those provided by your carrier.

- To enable **Override DNS**, check the **Override returned DNS Servers** box. Enter the IP addresses of the DNS servers into the spaces provided.



Override DNS

Override returned DNS Servers □ Use the following DNS Servers instead of the PPP provided servers.

DNS Server 1 — The Primary DNS Server.

DNS Server 2 — The Secondary DNS Server.

Dynamic DNS

Dynamic DNS: None - DDNS disabled — Update a DNS server when IP address is changed.

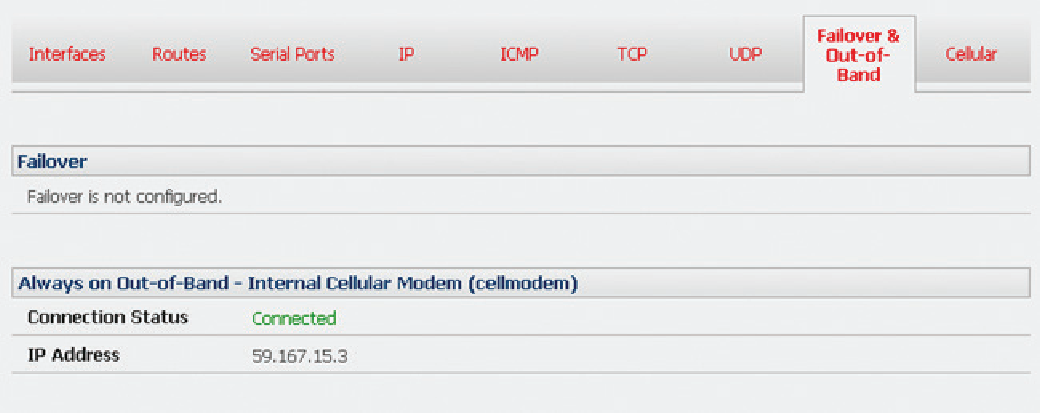
DDNS Hostname — The Fully Qualified DNS hostname assigned to this interface.

- Check **Apply** to establish a radio connection with your cellular carrier.

5.6.2 Verifying the cellular connection

Out-of-band access is enabled by default and the cellular modem connection should be established.

- You can verify the connection status from the **Status: Statistics** screen:
  - o Select the **Cellular** tab. When in *Service Availability*, verify that *Mode* is set to *Online*.
  - o Select **Failover & Out-of-Band**. The *Connection Status* will read *Connected*.
  - o Check your allocated *IP address*:



Failover

Failover is not configured.

Always on Out-of-Band - Internal Cellular Modem (cellmodem)

Connection Status: Connected

IP Address: 59.167.15.3

- You can measure the received signal strength from the Cellular Statistics page on the Status: Statistics screen. This will display the current state of the cellular modem, including the Received Signal Strength Indicator (RSSI)

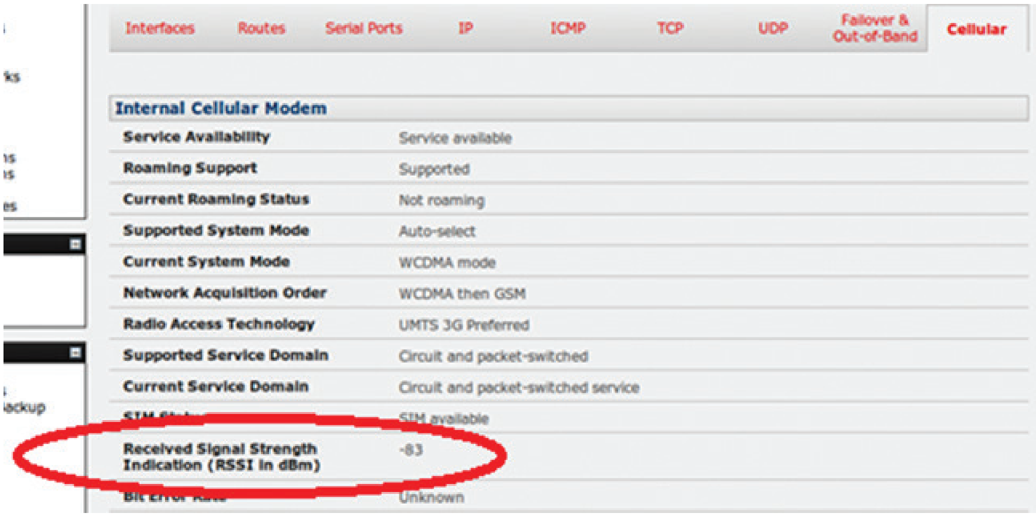
Note: *Received Signal Strength Indicator (RSSI) is a measurement of the Radio Frequency (RF) power present in a received radio signal on a mobile device. It is expressed in Decibel-milliwatts (dBm). The best throughput will result in placing the device in an area with the highest RSSI.*

-100 dbm or less = Unacceptable coverage

-99 dbm to -90 dbm = Weak Coverage

-89 dbm to -70 dbm = Medium to High Coverage

-69 dbm or greater = Strong Coverage

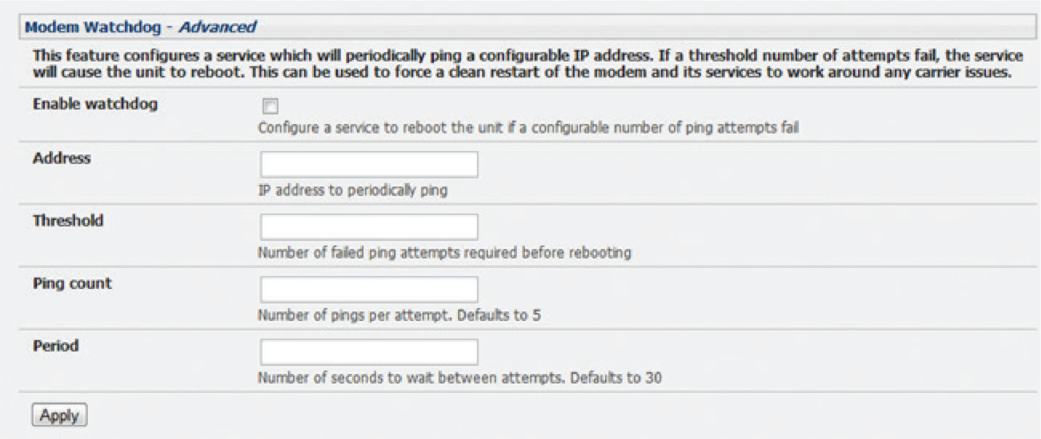


Internal Cellular Modem	
Service Availability	Service available
Roaming Support	Supported
Current Roaming Status	Not roaming
Supported System Mode	Auto-select
Current System Mode	WCDMA mode
Network Acquisition Order	WCDMA then GSM
Radio Access Technology	UMTS 3G Preferred
Supported Service Domain	Circuit and packet-switched
Current Service Domain	Circuit and packet-switched service
SIM Status	SIM available
Received Signal Strength Indication (RSSI in dBm)	-83
Bit Error Rate	Unknown

- With the cellular modem connection on, you can also check the connection status from the LEDs located on top of unit.

5.6.3 Cellular modem watchdog

When you select **Enable Dial-Out** on the **System: Dial** menu, 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 create a workaround for any carrier issues.



Modem Watchdog - *Advanced*

This feature configures a service which will periodically ping a configurable IP address. If a threshold number of 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.

Enable watchdog	□ Configure a service to reboot the unit if a configurable number of ping attempts fail
Address	IP address to periodically ping
Threshold	Number of failed ping attempts required before rebooting
Ping count	Number of pings per attempt. Defaults to 5
Period	Number of seconds to wait between attempts. Defaults to 30

5.7 Cellular Operation

When set up as a *console server*, the cellular modem can be set up to connect to the carrier in one of three modes:

- *Cellular router mode* – In this case, the dial-out connection to the carrier's cellular network is always on and IP traffic is routed between the cellular connected network and the console server's local network ports. This is the default mode of operation.
- *OOB mode* – The dial-out connection to the carrier's cellular network is always on and awaiting any incoming access (from a remote site seeking access to the console server or attached serial consoles/network hosts).
- *Failover mode* – A dial-out cellular connection is established only in the event of a *ping* failure.

5.7.1 OOB access set up

In this mode, the dial-out connection to the carrier's cellular network is always on and awaiting any incoming traffic. By default, the only traffic enabled are 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 transmitted over the cellular network. However, the status reports and site alerts are generally transmitted over the main network.

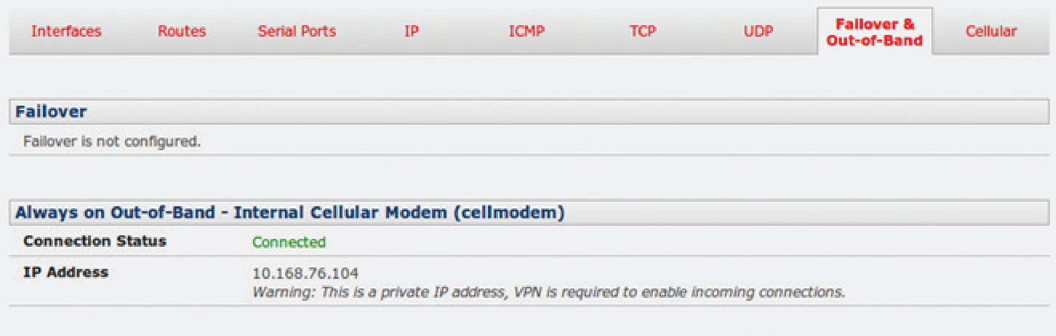
OOB mode is typically used for out-of-band access to remote sites by directly accessed appliances requiring a public IP address. OOB mode is the default for IM7200 and IM4200 appliances with internal cellular modems. Out-of-band access is enabled by default and the cellular modem connection is always on. For direct access, the *console server* requires a public IP address and must not have SSH access firewalled.

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

- If you have a static public IP address plan, you can also 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 (e.g., you can browse to the *console server*, but cannot *ping* it).
- If you have a dynamic public IP address plan, a DDNS service will need to be configured to enable the remote administrator to initiate incoming access. Once this is done, you can try accessing the *console server* using the allocated domain name.

By default, most providers offer a consumer-grade service that delivers dynamic private IP address assignments to cellular devices. This IP address is not visible across the Internet, but is generally adequate for home and general business use.

- With a consumer-grade plan, the **Failover & Out-of-Band** tab on the **Status: Statistics** will display that your carrier has allocated a *private IP Address* (i.e., within the range 10.0.0.0 – 10.255.255.255, 172.16.0.0 – 172.31.255.255 or 192.168.0.0 – 192.168.255.255).



Failover

Failover is not configured.

Always on Out-of-Band - Internal Cellular Modem (cellmodem)

Connection Status	Connected
IP Address	10.168.76.104 *Warning: This is a private IP address, VPN is required to enable incoming connections.*

- For an inbound OOB connection with a consumer-grade plan, you will need to set up an outbound VPN.

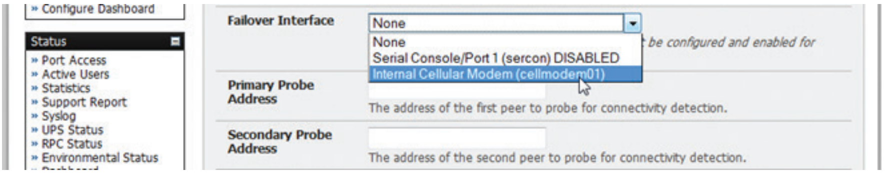
During out-of-band access mode, the internal cellular modem will continually stay connected. The alternative is to set up *Failover* mode on the *console server* (as detailed in the following section).

5.7.2 Cellular failover setup

In this mode, a dial-out cellular connection is established only when the main network is disrupted. The cellular connection will remain idle in a low power state and will only be activated in the event of a ping failure. This standby mode is well suited for remote sites with expensive power or extremely high cellular traffic costs.

In *Cellular failover startup* mode, the appliance will continually *ping* nominated probe addresses over the main network connection. In the event of *ping* failure, the appliance dials out and sets up a dial-out *ppp* over the cellular modem. Access is then switched to this network connection transparently. Access is switched back when the main network connection is restored. Once the carrier connection has been configured, the cellular modem can be configured for failover.

During *Cellular failover setup* mode, the cellular connection will remain idle and in a low power state. If the primary and secondary probe addresses are not available, it will reactivate the cellular network connection and reconnect with the cellular carrier.



Failover Interface: None

be configured and enabled for

Primary Probe Address

The address of the first peer to probe for connectivity detection.

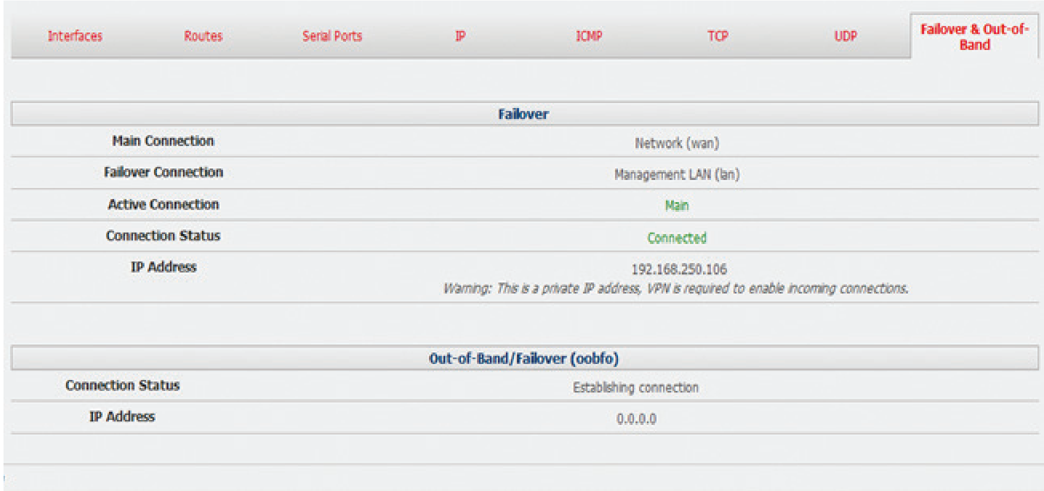
Secondary Probe Address

The address of the second peer to probe for connectivity detection.

- Navigate back to the **Network Interface** on the **System: IP** menu and specify **Internal Cellular modem (cell modem 01)** as the **Failover Interface** to be used when a fault has been detected.
- Specify the **Probe Addresses** of two sites **(Primary and Secondary)** that the *console server* is to *ping* to determine if the principal network is still operational.
- In the event of a principal network failure, the cellular network connection is activated as the access path to the console server (and Managed Devices). Only HTTPS and SSH access is enabled on the failover connection (doing this should enable the administrator to connect and fix the problem).

Note: *By default, the* console server *supports automatic failure-recovery back to the original state prior to failover. The* console server *continually pings probe addresses throughout 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.*

- You can check the connection status by selecting the C*ellular* panel on the **Status: Statistics** menu.



Failover	
Main Connection	Network (wan)
Failover Connection	Management LAN (lan)
Active Connection	Main
Connection Status	Connected
IP Address	192.168.250.106 *Warning: This is a private IP address, VPN is required to enable incoming connections.*

Out-of-Band/Failover (oobfo)	
Connection Status	Establishing connection
IP Address	0.0.0.0

- o The *Operational Status* will change as the cellular modem finds a channel and connects to the network.
- o The **Failover & Out-of-Band** screen displays information relating to a configured Failover/OOB interface and the status of that connection. The IP Address of the Failover/ OOB interface will be presented in the **Failover & Out-of-Band** screen once the Failover/OOB interface has been triggered.

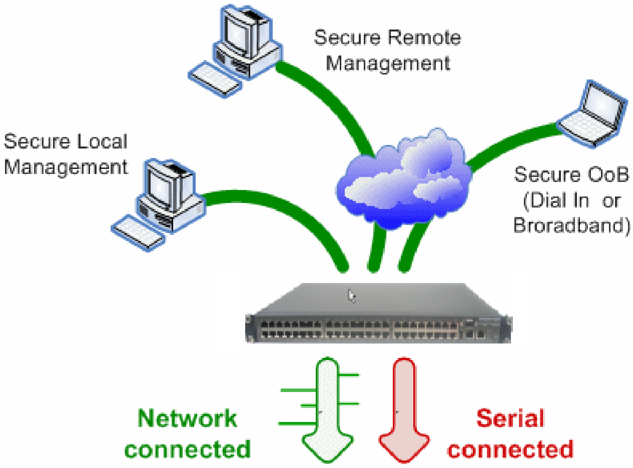
5.7.3 Cellular routing

Once you have configured your carrier connection, the cellular modem can be configured to route traffic through the *console server*. This requires setting up *forwarding* and *masquerading* firewall rules as detailed in Chapter 5.

Chapter 6: Secure SSH Tunneling & SDT Connector

Each Console Server has an embedded SSH server and uses SSH tunneling. This enables one Console Server to securely manage all the systems and network devices in the data center, using text-based console tools (such as SSH, Telnet, SoL) or graphical desktop tools (VNC, RDP, HTTPS, HTTP, X11, VMware, DRAC, iLO etc).

To set up Secure Tunnel access, the computer being accessed can be located on the same local network as the Console Server, or attached to the Console Server via its serial COM port. The remote User/Administrator then connects to the Console Server through an SSH tunnel (via dial-up, wireless or ISDN modem); a broadband Internet connection; an enterprise VPN network or a local network.



To set up the secure SSH tunnel from the Client computer to the Console Server, you must install and launch SSH client software on the User/Administrator's computer. It is recommended that you use the *SDT Connector* client software supplied with the Console Server to do this. *SDT Connector* is simple to install and it auto-configures. It provides 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 and then establishes a port forward connection to the target network connected host or serial connected device. It will then execute 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 Users access (*Section 6.1*)
- Setting up the SDT Connector client with gateway, host, service and client application details and making connections between the Client computer and hosts connected to the Console Server (*Section 6.2*)
- Using SDT Connector to browser access the Management Console (*Section 6.3*)
- Using SDT Connector to Telnet or SSH connect to devices that are serially attached to the Console Server (*Section 6.4*)

The chapter then covers more advanced SDT Connector and SDT tunneling topics:

- Using SDT Connector for out of band access (*Section 6.5*)
- Automatic importing and exporting of configurations (*Section 6.6*)
- Configuring Public Key Authentication (*Section 6.7*)
- Setting up a SDT Secure Tunnel for Remote Desktop (*Section 6.8*)
- Setting up a SDT Secure Tunnel for VNC (*Section 6.9*)
- Using SDT to IP connect to hosts that are serially attached to the Console Server (*Section 6.10*)

Chapter 6: Secure SSH Tunneling & SDT Connector

6.1 Configuring for SDT Tunneling to Hosts

To set up the Console Server to SDT access a network attached *host*, the *host* and the permitted *services* that are to be used in accessing that host need to be configured on the gateway, and User access privileges need to be specified:

- Add the new *host* and the *permitted services* using the **Serial & Network: Network Hosts** menu as detailed in *Network Hosts* (*Chapter 4.4*). Only these *permitted services* will be forwarded by SDT to the *host*. All other services (TCP/UDP ports) will be blocked.

***Note**: Following are some of the TCP Ports used by SDT in the Console Server:*

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 – where XX is the serial port number (i.e. 7301to 7348)*
79XX	*VNC over serial from local LAN – where XX is the serial port number*

- Add the new *Users* using **Serial & Network: Users & Groups** menu as detailed in *Network Hosts* (*Chapter 4.4*). Users can be authorized to access the Console Server ports and specified network-attached hosts. To simplify configuration, the Administrator can first set up *Groups* with group access permissions, then Users can be classified as members of particular *Groups*.

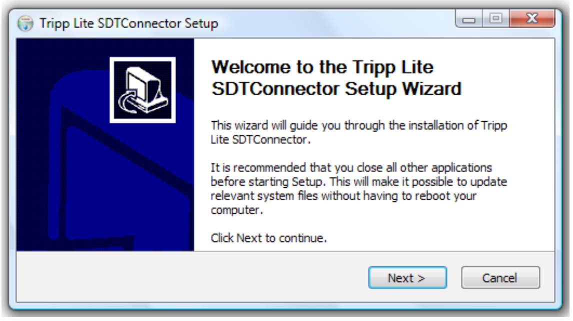
6.2 SDT Connector Configuration

The *SDT Connector* client works with all Console Servers. Each of these remote Console Servers has an embedded OpenSSH based server. This server can be configured to *port forward* connections from the *SDT Connector* client to hosts on their local network, as detailed in the previous chapter. The *SDT Connector* can also be pre-configured with the access tools and applications that will be available 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 be configured to access the Console Server itself and to access devices connected to serial ports on the Console Server.

6.2.1 SDT Connector client installation

- The *SDT Connector* set up program (**SDTConnector Setup-1.n.exe** or **sdtcon-1.n.tar.gz**) is included on the CD supplied with your Console Server
- Run the set-up program:



Welcome to the Tripp Lite SDTConnector Setup Wizard

This wizard will guide you through the installation of Tripp Lite SDTConnector.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

***Note**: For Windows clients, the **SDTConnectorSetup-1.n.exe** application will install the **SDT Connector 1.n.exe** and the config file **defaults.xml**. If a config file already exists on the Windows computer, then it will not be overwritten. To remove an earlier config file, run the **regedit** command, search for "SDT Connector" and then remove the directory with this name.*

*For Linux and other Unix clients, **SDTConnector.tar.gz** application will install the **sdtcon-1.n.jar** and the config file **defaults.xml***

Once the installer completes, you will have a working SDT Connector client installed on your machine and an icon on your desktop:

- Click the *SDT Connector* icon on your desktop to start the client

***Note**: SDT Connector is a Java application so it must have a Java Runtime Environment (JRE) installed. This can be freely downloaded from http://java.sun.com/j2se/ . It will install on Windows 2000, XP, 2003, Vista computers 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*

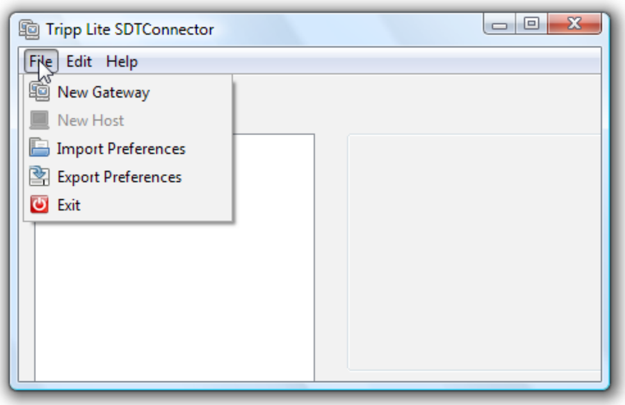
To operate *SDT Connector*, add the new gateways to the client software by entering the access details for each Console Server (refer to Section 6.2.2). Then let the client auto-configure with all host and serial port connections from each Console Server (refer *Section 6.2.3*). Now point-and-click to connect to the Hosts and serial devices (refer to *Section 6.2.4*)

Alternately you can manually add network connected hosts (refer *Section 6.2.5*) as well as manually configure new services to be used when accessing the Console Server and the hosts (refer *Section 6.2.6*). Manually configure clients to run on the computer that will use the service to connect to the hosts and serial port devices (refer to *Section 6.2.7 and 6.2.9*). *SDT Connector* can also be set up to make an out-of-band connection to the Console Server (refer to *Section 6.2.9*)

6.2.2 Configuring a new gateway in the SDT Connector client

To create a secure SSH tunnel to a new Console Server:

- Click the New Gateway icon or select the **File: New Gateway** menu option

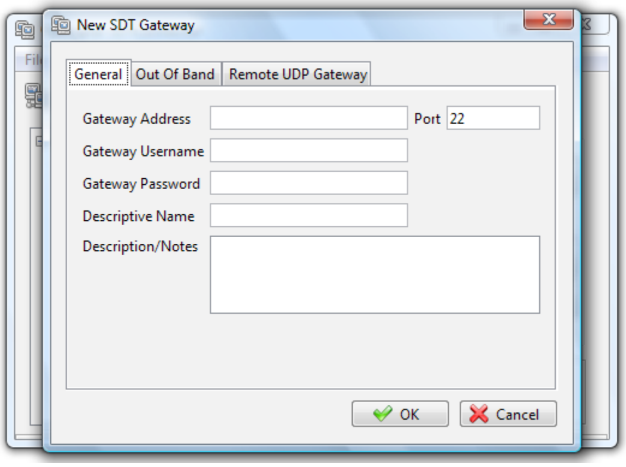


- Enter the IP or DNS **Address** of the Console Server and the SSH port that will be used (typically 22)

***Note**: If **SDT Connector** is connecting to a remote Console Server through the public Internet or routed network, you will need to:*

- *Determine **the public IP address** of the Console Server (or of the router/ firewall that connects the Console Server to the Internet) as assigned by the ISP. One way to find the public IP address is to access http://checkip.dyndns.org/ or http://www.whatismyip.com/ from a computer on the same network as the Console Server and note the reported IP address*
- *Set port forwarding for TCP port 22 through any firewall/NAT/router that is located between **SDT Connector** and the Console Server so that it points to the Console Server. http://www.portforward.com has port forwarding instructions for a range of routers. Also you can use the Open Port Check tool from http://www.canyouseeme.org to check if port forwarding through local firewall/NAT/router devices has been properly configured*

- Enter the **Username** and **Password** of a user on the gateway that has been enabled to connect via SSH and/or create SSH port redirections



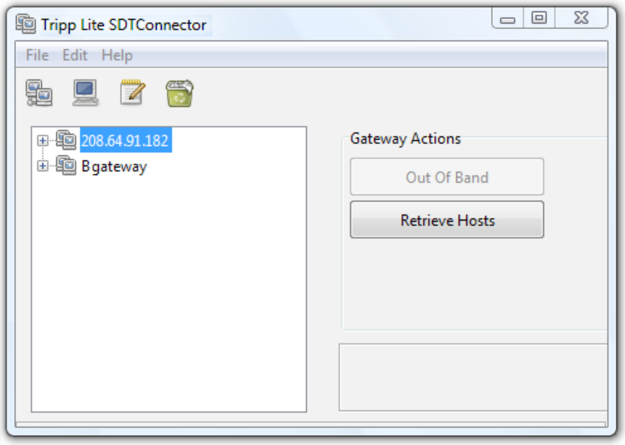
Port 22

- Optionally, you can enter a **Descriptive Name** to display instead of the IP or DNS address, and any **Notes** or a **Description** of this gateway (such as its firmware version, site location or anything special about its network configuration).
- Click **OK** and an icon for the new gateway will now appear in the *SDT Connector* home page

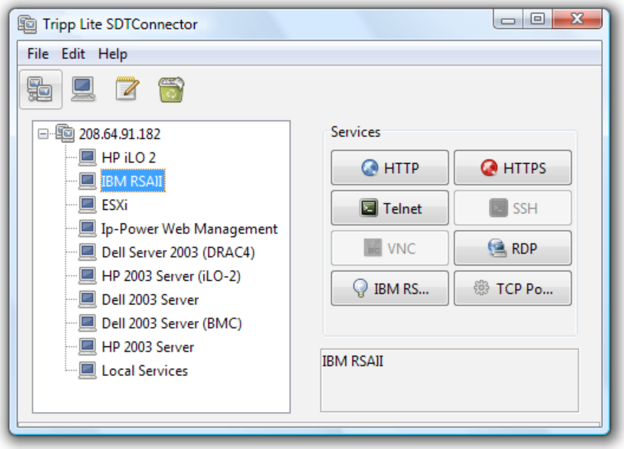
***Note**: For an **SDT Connector** user to access a Console Server (and then access specific hosts or serial devices connected to that Console Server), that user must first be set up on the Console Server, and must be authorized to access the specific ports / hosts (refer to Chapter 5). Only these **permitted services** will be forwarded through by SDT to the Host. All other services (TCP/UDP ports) will be blocked.*

6.2.3 Auto-configure SDT Connector client with the user's access privileges

Each user on the Console Server has an access profile. This has been configured with the specific connected hosts and serial port devices the user has authority to access, and a specific set of the enabled services for each of them. This configuration can be auto-uploaded into the SDT Connector client:



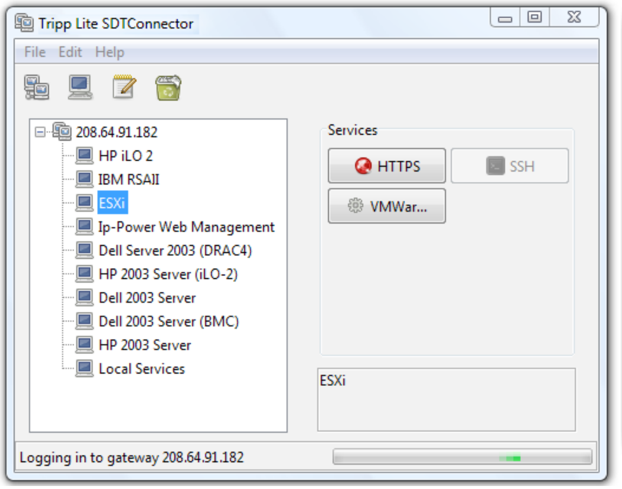
- Click on the new gateway icon and select **Retrieve Hosts**. This will:
  - configure access to network-connected Hosts that the user is authorized to access and set up (for each of these Hosts) the services (e.g. HTTPS, IPMI2.0) and the related IP ports being redirected
  - configure access to the 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



***Note**: The Retrieve Hosts function will auto-configure all classes of user (i.e. they can be members of **user** or **admin** or some other group or no group). SDT Connector will, however, not auto-configure the **root** (and it is recommended that this account is only used for initial config and for adding an initial **admin** account to the Console Server)*

6.2.4 Make an SDT connection through the gateway to a host

- Simply ***point*** at the host to be accessed ***and click*** on the service to be used in accessing that host. The SSH tunnel to the gateway is then automatically 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:



Logging in to gateway 208.64.91.182

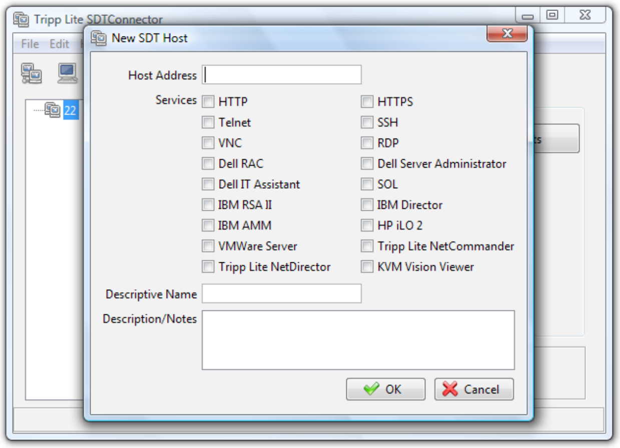
Note*: The SDT Connector client can be configured with an unlimited number of Gateways. Each Gateway can be configured to port forward to an unlimited number of locally networked Hosts. Similarly there is no limit on the number of SDT Connector clients who can be configured to access the one Gateway. There are also no limits on the number of Host connections that an SDT Connector client can concurrently have open through the one Gateway tunnel.*

However, there is a limit on the number of SDT Connector SSH tunnels that can be open at one time on a particular Gateway. The B096-016 / B096-032 / B096-048 Console Server Management Switch and B092-016 Console Server with PowerAlert each support at least 50 such concurrent connections. So for a site with a B096-016 gateway you can have, at any time, up to 50 users securely controlling an unlimited number of network attached computers, power devices and other appliances (routers, etc) at that site.

6.2.5 Manually adding hosts 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 the newly added gateway and click the *Host* icon to create a host that will be accessible via this gateway. (Alternatively select **File: New Host**)



New SDT Host

Host Address

Services	
□ HTTP	□ HTTPS
□ Telnet	□ SSH
□ VNC	□ RDP
□ Dell RAC	□ Dell Server Administrator
□ Dell IT Assistant	□ SOL
□ IBM RSA II	□ IBM Director
□ IBM AMM	□ HP iLO 2
□ VMWare Server	□ Tripp Lite NetCommander
□ Tripp Lite NetDirector	□ KVM Vision Viewer

Descriptive Name

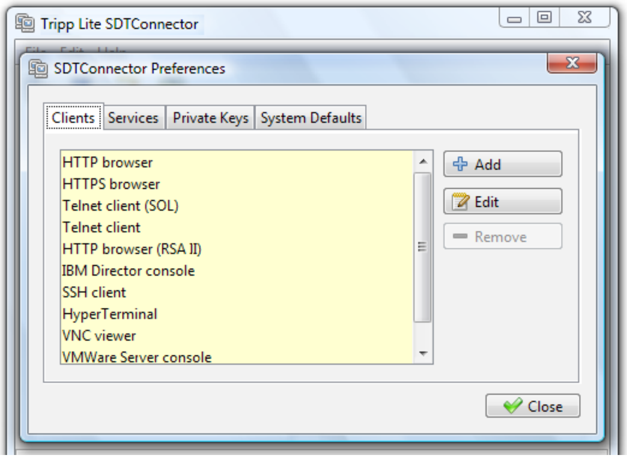
Description/Notes

- Enter the IP or DNS **Host Address** of the host (if this is a DNS address, it must be resolvable by the gateway)
- Select which **Services** are to be used when 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). However if you wish to add new services to the range then proceed to the next section (**Adding a new service**) then return here
- Optionally, you can enter a **Descriptive Name** for the host to be displayed instead of the IP or DNS address, as well as any **Notes** or a **Description** of this host (such as its operating system/release, or anything special about its configuration)
- Click **OK**

6.2.6 Manually adding new services to the new hosts

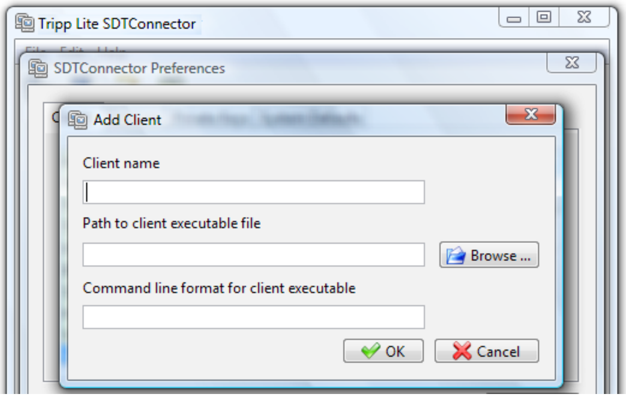
To extend the range of services that can be used when accessing hosts with *SDT Connector*:

- Select **Edit: Preferences** and click the **Services** tab. Click **Add**
- Enter a **Service Name** and click **Add**
- Under the **General** tab, enter the TCP Port that this service runs on (e.g. 80 for HTTP). Optionally, select the client to be used to access the local endpoint of the redirection



HTTP browser
HTTPS browser
Telnet client (SOL)
Telnet client
HTTP browser (RSA II)
IBM Director console
SSH client
HyperTerminal
VNC viewer
VMWare Server console

- 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). However if you wish to add new client applications to this range, then proceed to the next section (**Adding a new client**) and then return here

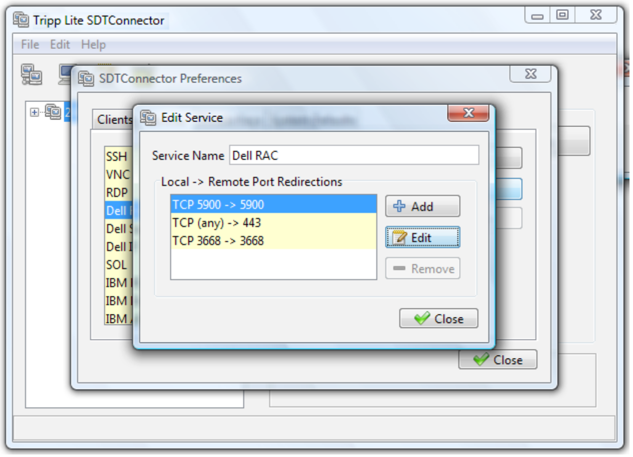


- Click **OK**, then **Close**

A service typically consists of a single SSH port redirection and a local client to access it. However 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 launch later 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.



Service Name: Dell RAC

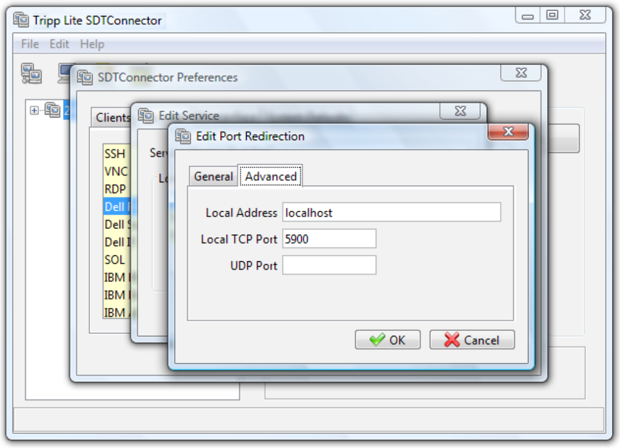
Local -> Remote Port Redirections:

TCP 5900 -> 5900
TCP (any) -> 443
TCP 3668 -> 3668

- On the Add Service screen, you can click **Add** as many times as needed to add multiple new port redirections and associated clients

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.



Local Address: localhost
Local TCP Port: 5900
UDP Port:

***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.*

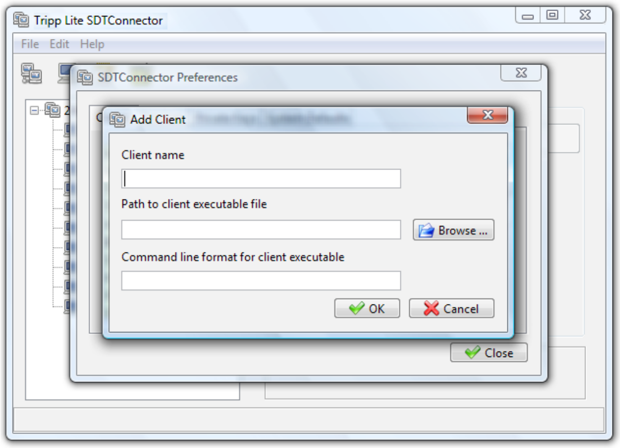
*Enter the UDP port on which the service is running on the host. This will also be the local UDP port that **SDT Connector** binds as the local endpoint of the tunnel.*

Note that 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

6.2.7 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 and click the **Client** tab. Click **Add**



- Enter a **Name** for the client. Enter the **Path** to the executable file for the client (or click **Browse** to locate the executable)

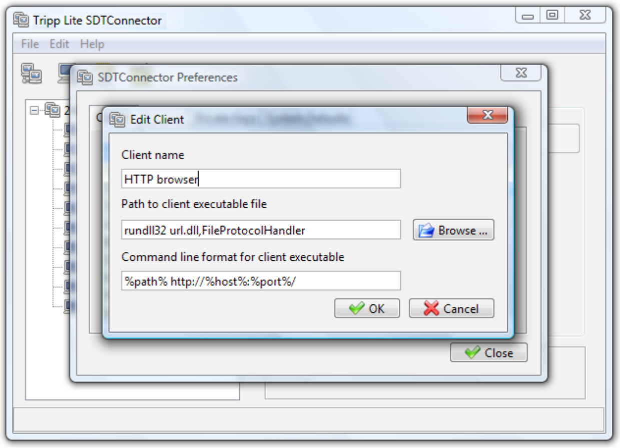
- Enter a **Command Line** associated with launching the client application. *SDT Connector* typically launches a client using command line arguments to point it to the local endpoint of the redirection. There are three special keywords for specifying the command line format. When launching the client, *SDT Connector* substitutes these keywords with the appropriate values:

  %path% is path to the executable file, i.e. the previous field.

  %host% is the local address to which the local endpoint of the redirection is bound, i.e. the Local Address field for the Service redirection Advanced options.

  %port% is the local port to which the local endpoint of the redirection is bound, i.e. the Local TCP Port field for the Service redirection Advanced options. If this port is unspecified (i.e. "Any"), the appropriate randomly selected port will be substituted.

For example, *SDT Connector* is preconfigured for Windows installations with a HTTP service client that will connect with whichever local browser the local Windows user has configured as the default. Otherwise the default browser used is Firefox:

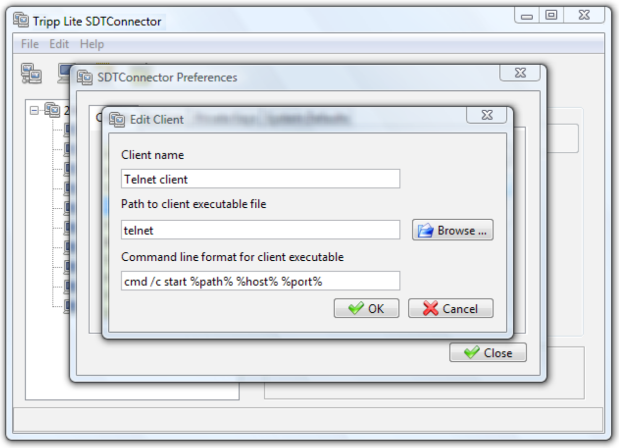


Client name: HTTP browser

Path to client executable file: rundll32 url.dll,FileProtocolHandler

Command line format for client executable: %path% http://%host%:%port%/

Also some clients are launched in a command line or terminal window. The Telnet client is an example of this:



Client name: Telnet client

Path to client executable file: telnet

Command line format for client executable: cmd /c start %path% %host% %port%

- Click OK

6.2.8 Dial-in configuration

If the client computer is dialing into Local/Console port on the Console Server, you will need to set up a dial-in PPP link:

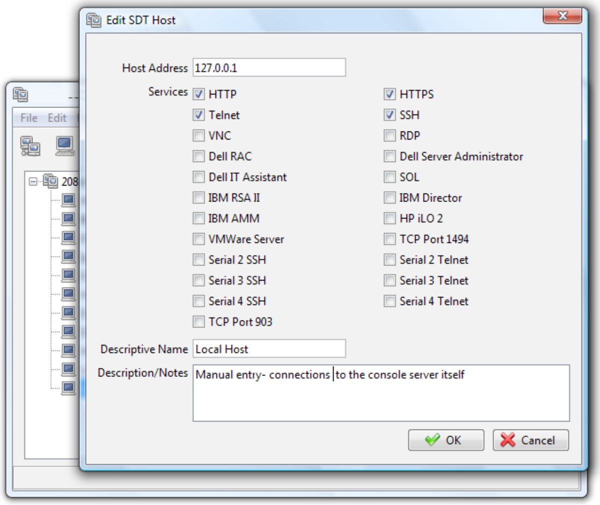
- Configure the Console Server for dial-in access (following the steps in the ***Configuring for Dial-In PPP Access*** section in *Chapter 5, Configuring Dial In Access*)
- Set up the PPP client software at the remote User computer (following the ***Set up the remote Client*** section in *Chapter 5*)

Once you have a dial-in PPP connection established, you can then set up the secure SSH tunnel from the remote Client computer to the Console Server.

6.3 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 command line. For these connections to the gateway itself, you must configure *SDT Connector* to access the gateway (itself) by setting the Console Server up as a *host*, and then configuring the appropriate services:

- Launch *SDT Connector* on your computer. 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** and give some details in **Descriptive Name/Notes**. Click OK



Edit SDT Host

Host Address 127.0.0.1

Services

☑ HTTP	☑ HTTPS
☑ Telnet	☑ SSH
☐ VNC	☐ RDP
☐ Dell RAC	☐ Dell Server Administrator
☐ Dell IT Assistant	☐ SOL
☐ IBM RSA II	☐ IBM Director
☐ IBM AMM	☐ HP iLO 2
☐ VMWare Server	☐ TCP Port 1494
☐ Serial 2 SSH	☐ Serial 2 Telnet
☐ Serial 3 SSH	☐ Serial 3 Telnet
☐ Serial 4 SSH	☐ Serial 4 Telnet
☐ TCP Port 903	

Descriptive Name Local Host

Description/Notes Manual entry- connections to the console server itself

- Click the **HTTP** or **HTTPS** Services icon to access the gateway's Management Console, and/or click **SSH** or **Telnet** to access the gateway command line console

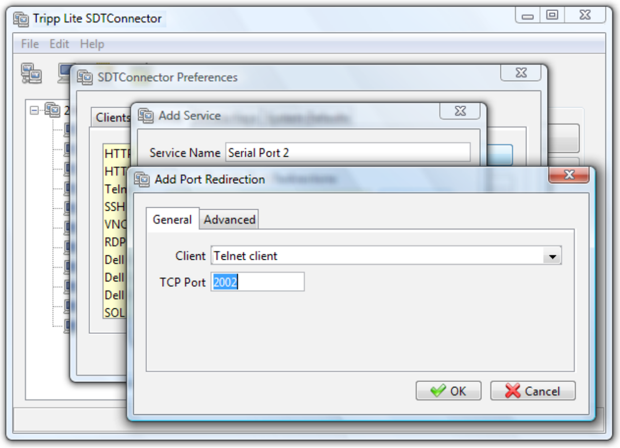
Note*: To enable SDT access to the gateway console, you must now configure the Console Server to allow port forwarded network access to itself:*

- *Browse to the Console Server and select **Network Hosts** from **Serial & Network**. Click **Add Host** and in the **IP Address/ DNS Name** field enter 127.0.0.1 (this is the Console Server's network loopback address). Then enter **Loopback** in **Description***
- *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). Scroll to the bottom and click **Apply***
- *Administrators by default have gateway access privileges. However for Users to access the gateway Management Console, you will need to give those Users the required access privileges. Select **Users & Groups** from **Serial & Network**. Click **Add User**. Enter a **Username, Description** and **Password/Confirm**. Select 127.0.0.1 from **Accessible Host(s)** and click **Apply***

6.4 SDT Connector - Telnet or SSH Serial Device Connection

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, you must 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 computer. Select **Edit: Preferences** and click the **Services** tab. Click **Add**
- Enter *"Serial Port 2"* in **Service Name** and click **Add**
- Select **Telnet** client as the Client. Enter 2002 in **TCP Port**. Click **OK**, then **Close** and **Close** again



Service Name: Serial Port 2

Client: Telnet client

TCP Port: 2002

- Assuming you have already set up the target Console Server as a gateway in your *SDT Connector* client (with *username/ password* etc), select this *gateway* and click the **Host** icon to create a host. Alternatively, select **File: New Host**.
- Enter 127.0.0.1 as the **Host Address** and select **Serial Port 2** for Service. In **Descriptive Name**, enter something along the lines of Loopback ports, or Local serial ports. Click **OK**.
- Click *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, you must also configure the Console Server itself to allow port forwarded network access to itself, and enable access to the nominated serial port:

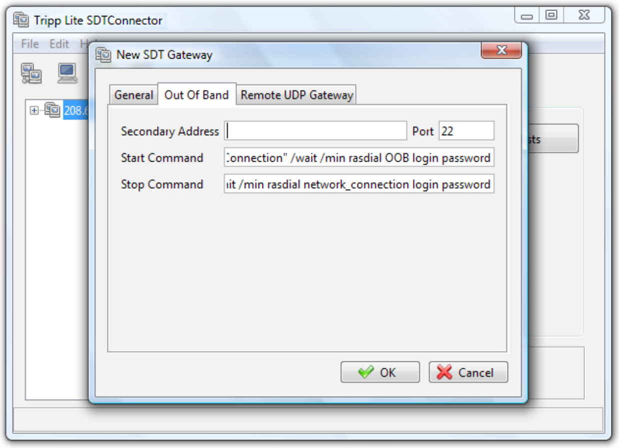
- Browse to the Console Server and select **Serial Port** from **Serial & Network**
- Click **Edit** to selected Port # (e.g. Port 2 if the target device is attached to the second serial port). Ensure the port's serial configuration is appropriate for the attached device
- Scroll down to **Console Server Setting** and select **Console Server Mode**. Check **Telnet** (or SSH) and scroll to the bottom and click **Apply**
- Select **Network Hosts** from **Serial & Network** and click **Add Host**
- In the **IP Address/DNS Name** field, enter *127.0.0.1* (this is the Console Server's network loopback address) and enter *Loopback* in **Description**
- Remove all entries under **Permitted Services** and select **TCP** and enter *200n* in **Port**. (This configures the Telnet port enabled in the previous step, so for Port 2 you would enter *2002*)
- Click **Add** then scroll to the bottom and click **Apply**
- Administrators by default have gateway and serial port access privileges; however for Users to access the gateway and the serial port, you will need to give those Users the required access privileges. Select **Users & Groups** from **Serial & Network**. Click **Add User**. Enter a **Username, Description** and **Password/Confirm**. Select 127.0.0.1 from **Accessible Host(s)** and select Port 2 from Accessible Port(s). Click **Apply**.

6.5 SDT Connector OoB Connection

SDT Connector can also be set up to connect to the Console Server via out-of-band (OoB). OoB access uses an alternate path for connecting to the Console Server (i.e. not the one used for regular data traffic). OoB access is useful when the primary link into the gateway 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 gateway. So out-of-band access enables you to access the hosts and serial devices on the network, diagnose any connectivity issues, and restore the gateway'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 gateway. *SDT Connector* allows for maximum flexibility by allowing you to provide your own scripts or commands for starting and stopping the OoB connection.



Secondary Address
Port 22
Start Command :onnection" /wait /min rasdial OOB login password
Stop Command ıit /min rasdial network_connection login password

To configure *SDT Connector* for OoB access:

- When adding a new gateway or editing an existing gateway, select the **Out Of Band** tab
- Enter the secondary OoB IP address for the gateway (e.g. the IP address to be used when dialing in directly). You may also modify the gateway's SSH port if it's not using the default of 22
- Enter the command or path to a script to start the OoB connection in **Start Command**
    - o To initiate a pre-configured dial-up connection under Windows, use the following Start Command:

      cmd /c start "Starting Out of Band Connection" /wait /min rasdial network_connection login password

      The *network_connection* in the above is the name of the network connection as displayed in *Control Panel -> Network Connections*. *Login* is the dial-in username, and *password* is the dial-in password for the connection.
    - o To initiate a pre-configured dial-up connection under Linux, use the following Start Command:

      pon network_connection

      The *network_connection* in *the above is the name of the connection.*
- Enter the command or path to a script to stop the OoB connection in **Stop Command**
    - o To stop a pre-configured dial-up connection under Windows, use the following Stop Command:

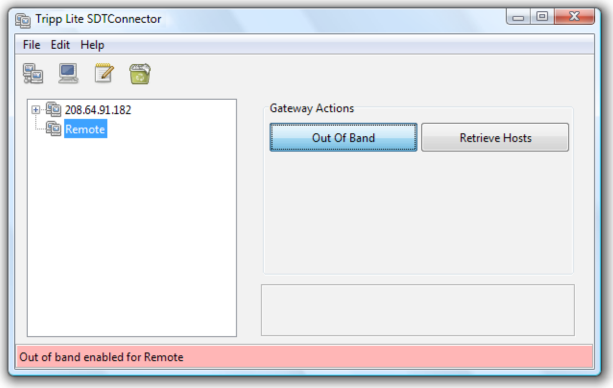
      cmd /c start "Stopping Out of Band Connection" /wait /min rasdial network_connection /disconnect

      The network_connection in the above is the name of the network connection as displayed in *Control Panel -> Network Connections*.
    - o To stop a pre-configured dial-up connection under Linux, use the following Stop Command:

      poff network_connection

To make the OoB connection using *SDT Connector*:

- Select the gateway and click Out Of Band. The status bar will change color to indicate this gateway is now being access using the OoB link rather than the primary link

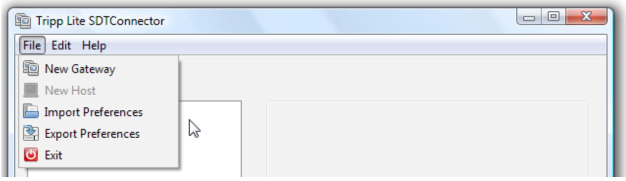


Out of band enabled for Remote

When you connect to a service on a host behind the gateway, or to the Console Server gateway 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.

6.6 Importing (and Exporting) Preferences

To enable the distribution of pre-configured client config files, *SDT Connector* has an *Export/Import* facility:



- To save a configuration .xml file (for backup or for importing into other *SDT Connector* clients), select **File: Export Preferences** and select the location to save the configuration file
- To import a configuration, select **File: Import Preferences** and select the .xml configuration file to be installed

6.7 SDT Connector Public Key Authentication

SDT Connector can authenticate against an SSH gateway using your SSH key pair rather than requiring your to enter your password. This is known as public key authentication.
To use public key authentication with SDT Connector, you must first 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 computer (the one which the SDT Connector is running) generate them now using *ssh-keygen, PuTTYgen* or a similar tool. You may use RSA or DSA, however it is important that you leave the passphrase field blank:

o	PuTTYgen:	http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
o	OpenSSH:	http://www.openssh.org/
o	OpenSSH (Windows):	http://sshwindows.sourceforge.net/download/

- Upload the public part of your *SSH* key pair (this file is typically named *id_rsa.pub* or *id_dsa.pub*) to the SSH gateway, or add it to the *.ssh/authorized keys* in your home directory on the SSH gateway
- Next, add the private part of your SSH key pair (this file is typically named *id_rsa or id_dsa*) to SDT Connector.
  Click **Edit: Preferences: Private Keys: Add**, locate the private key file and 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 connecting through the SSH gateway (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 may also wish to configure access to it for public key authentication as well. This configuration is entirely independent of SDT Connector and the SSH gateway. You must configure the SSH client that SDT Connector launches (e.g. Putty, 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.

6.8 Setting up SDT for Remote Desktop Access

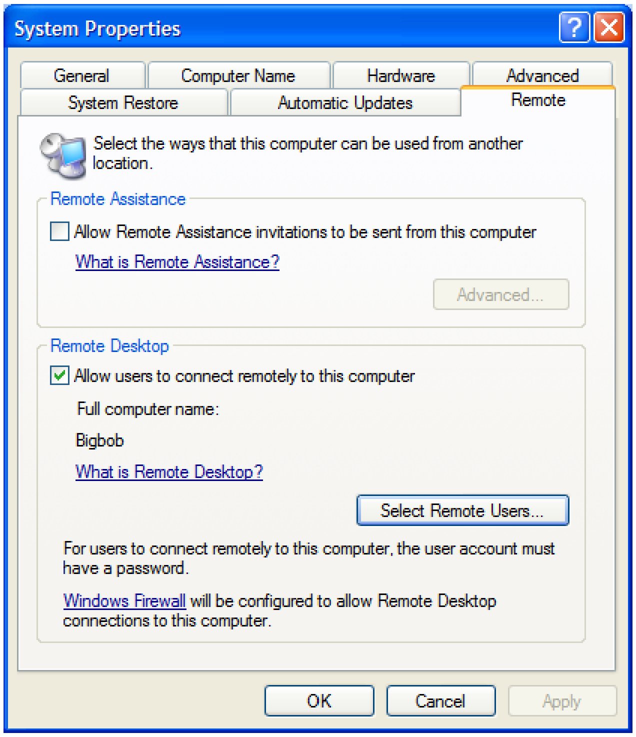
Microsoft's Remote Desktop Protocol (RDP) enables the system manager securely to access and manage remote Windows computers: to reconfigure applications and user profiles, upgrade the server's operating system, reboot the machine, etc. Secure Tunneling uses SSH tunneling, so this RDP traffic is securely transferred through an authenticated and encrypted tunnel.

SDT with RDP also allows remote Users to connect to Windows XP, Vista, Windows 2003 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 itself). 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 computer.

6.8.1 Enable Remote Desktop on the target Windows computer to be accessed

To enable **Remote Desktop** on the Windows computer being accessed:

- Open **System** in the Control Panel and click the **Remote** tab



Select the ways that this computer can be used from another location.

□ Allow Remote Assistance invitations to be sent from this computer

☑ Allow users to connect remotely to this computer

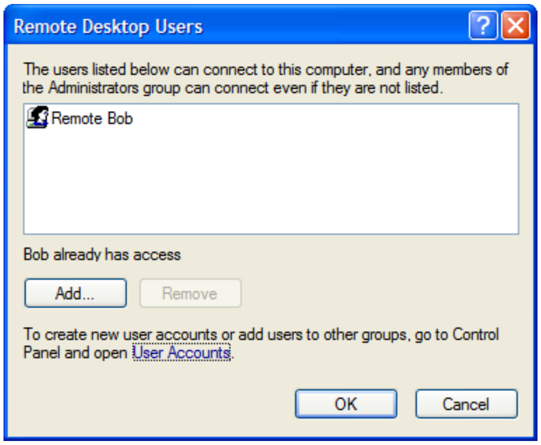
Full computer name:

Bigbob

For users to connect remotely to this computer, the user account must have a password.

Windows Firewall will be configured to allow Remote Desktop connections to this computer.

- Check **Allow users to connect remotely to this computer**
- Click **Select Remote Users**



Remote Desktop Users

The users listed below can connect to this computer, and any members of the Administrators group can connect even if they are not listed.

Remote Bob

Bob already has access

To create new user accounts or add users to other groups, go to Control Panel and open User Accounts.

- To set the user(s) who can remotely access the system with RDP, click Add on the Remote Desktop Users dialog box

***Note**: If you need to set up new users for Remote Desktop access, open **User Accounts** in the Control Panel and proceed through the steps to nominate the new user's name, password and account type (Administrator or Limited)*

***Note**: With Windows XP Professional and Vista, you have only one Remote Desktop session and it connects directly to the Windows root console. With Windows Server 2008 you can have multiple sessions, and with Server 2003 you have three sessions (the console session and two other general sessions). Therefore, more than one user can have an active session on a single computer.*

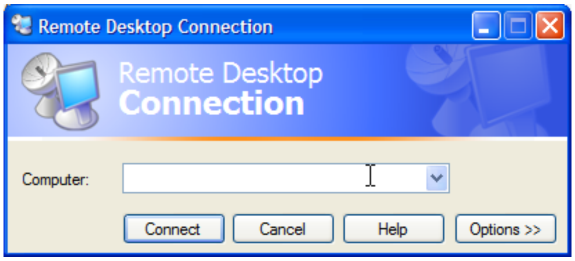
When the remote user connects to the accessed computer on the console session, Remote Desktop automatically locks that computer (so no other user can access the applications and files). When you come back to the computer, you can unlock it by typing CTRL+ALT+DEL.

6.8.2 Configure the Remote Desktop Connection client

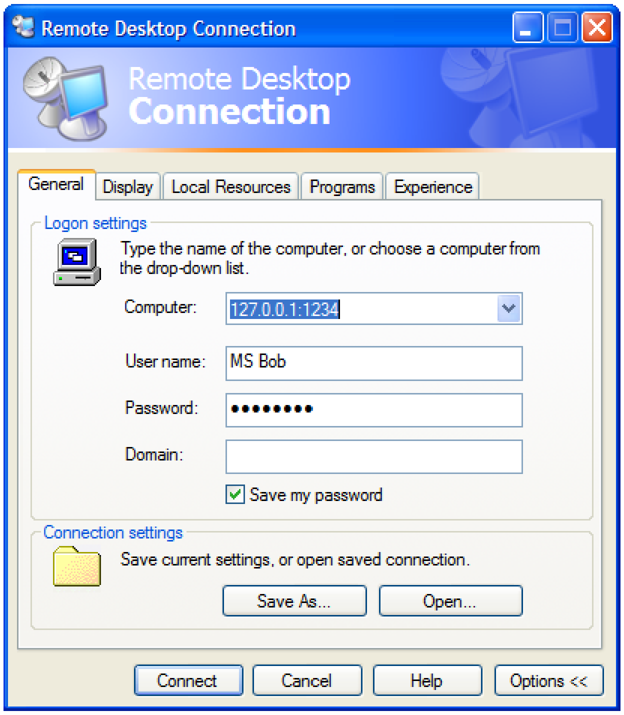
Now that you have the Client computer securely connected to the Console Server (either locally, or remotely, thru the enterprise VPN, or a secure SSH internet tunnel or a dial-in SSH tunnel), you are ready to establish the Remote Desktop connection from the Client. To do this you simply enable the Remote Desktop Connection on the remote client computer then point it to the SDT Secure Tunnel port in the Console Server:

A. On a Windows client computer

- Click **Start**. Point to **Programs**, then to **Accessories**, then **Communications**, and click **Remote Desktop Connection**



- In **Computer**, enter the appropriate IP Address and Port Number:
    - o Where there is a direct local 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's serial port (the one 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 then you would enter *192.168.0.50:7303*.
    - o Where there is an SSH tunnel (over a dial-up PPP connection or over a public internet connection or private network connection), simply enter the *localhost* as the IP address, i.e. *127.0.0.1*. For Port Number, enter the *source port* you created when setting SSH tunneling/port *forwarding (in Section 6.1.6) e.g.:1234.*
- Click **Option**. In the **Display** section, specify an appropriate color depth (e.g. for a modem connection it is recommended you not use over 256 colors). In **Local Resources**, specify the peripherals on the remote Windows computer that are to be controlled (printer, serial port, etc.)



Logon settings

Type the name of the computer, or choose a computer from the drop-down list.

Computer: 127.0.0.1:1234

User name: MS Bob

Password: ••••••••

Domain:

☑ Save my password

Connection settings

Save current settings, or open saved connection.

- Click Connect

Note*: The Remote Desktop Connection software is pre-installed on Windows XP. However, for earlier Windows computers, you will need to download the RDP client:*

- *Go to the Microsoft Download Center site http://www.microsoft.com/downloads/details.aspx?familyid=80111F21-D48D-426E-96C2-08AA2BD23A49&displaylang=en and click the* ***Download*** *button*

This software package will install the client portion of Remote Desktop on Windows 95, Windows 98 and 98 Second Edition, Windows Me, Windows NT 4.0, Windows 2000, and Windows 2003. When run, this software allows these older Windows platforms to remotely connect to a computer running Windows XP Professional or Windows 2003 Server

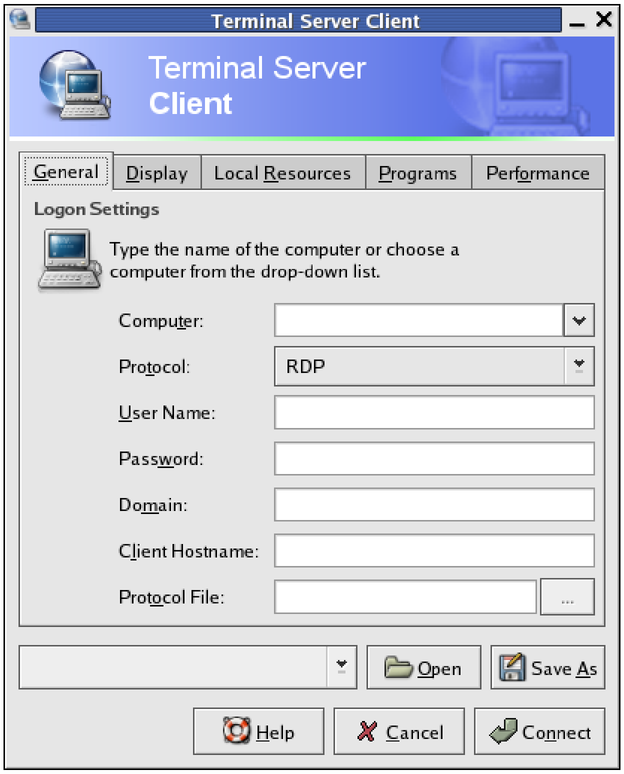
B. On a Linux or UNIX client computer:

- Launch the open source rdesktop client:

rdesktop -u *windows-user-id* -p *windows-password* -g 1200x950 *ms-windows-terminal-server-host-name*

option	description
-a	Color depth: 8, 16, 24
-r	Device redirection. i.e. Redirect sound on remote machine to local device i.e. -0 -r sound (MS/Windows 2003)
-g	Geometry: *widthxheight* or 70% screen percentage.
-p	Use -p - to receive password prompt.

- You can use GUI front end tools like the GNOME Terminal Services Client *tsclient* to configure and launch the *rdesktop* client. (Using *tsclient* also enables you to store multiple configurations of *rdesktop* for connection to many servers.)



Type the name of the computer or choose a computer from the drop-down list.

Protocol: RDP

***Note**: The rdesktop client is supplied with Red Hat 9.0:*

- *rpm -ivh rdesktop-1.2.0-1.i386.rpm*

For Red Hat 8.0 or other distributions of Linux; download source, untar, configure, make, make then install.

***rdesktop** currently runs on most UNIX based platforms with the X Window System and can be downloaded from http://www.rdesktop.org/*

C. On a Macintosh client:

- Download Microsoft's free Remote Desktop Connection client for Mac OS X http://www.microsoft.com/mac/otherproducts/otherproducts.aspx?pid=remotedesktopclient

6.9 SDT SSH Tunnel for VNC

Alternately, with SDT and Virtual Network Computing (VNC), Users and Administrators can securely access and control Windows 98/NT/2000/XP/2003, Linux, Macintosh, Solaris and UNIX computers. There's a range of popular VNC software available (UltraVNC, RealVNC, TightVNC) freely and commercially. To set up a secure VNC connection, install and configure the VNC Server software on the computer to be accessed. Then install and configure the VNC Viewer software on the Viewer computer.

6.9.1 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, Macintosh, Solaris, UNIX, all versions of Windows and most other operating systems.

A. For Microsoft Windows servers (and clients):

Windows does not include VNC software, so you will need to download, install and activate a third party VNC Server software package:

RealVNC http://www.realvnc.com is fully cross-platform, so a desktop running on a Linux machine may be displayed on a Windows computer, on a Solaris machine, or on any number of other architectures. There is a Windows server, allowing you to view the desktop of a remote Windows machine on any of these platforms using exactly the same viewer. RealVNC was founded by members of the AT&T team who originally developed VNC.

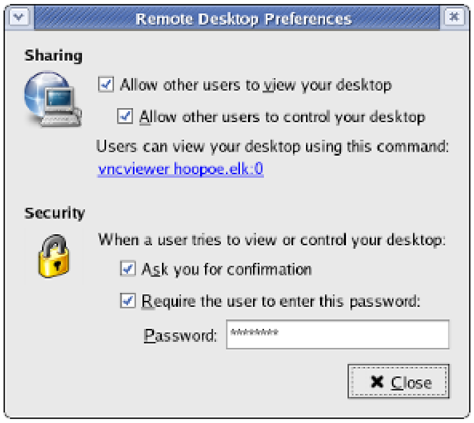
TightVNC http://www.tightvnc.com is an enhanced version of VNC. It has added features such as file transfer, performance improvements and read-only password support. They have just recently included a video drive much like UltraVNC. TightVNC is still free, cross-platform (Windows Unix and Linux) and compatible with the standard (Real) VNC.

UltraVNC http://ultravnc.com is easy to use, fast and free VNC software that has pioneered and perfected features that the other flavors have consistently refused or been very slow to implement for cross platform and minimalist reasons. UltraVNC runs under Windows operating systems (95, 98, Me, NT4, 2000, XP, 2003) Download UltraVNC from Sourceforge's UltraVNC file list

B. For Linux servers (and clients):

Most Linux distributions now include VNC Servers and Viewers. They are generally launched from the (Gnome/KDE etc) front end. For example, there's VNC Server software with Red Hat Enterprise Linux 4 and a choice of Viewer client software. To launch:

- Select the **Remote Desktop** entry in the **Main Menu: Preferences** menu
- Click the **Allow other users** checkbox to allow remote users to view and control your desktop


Remote Desktop Preferences

Sharing

☑ Allow other users to view your desktop

☑ Allow other users to control your desktop

Users can view your desktop using this command:
vncviewer hoopoe.elk:0

Security

When a user tries to view or control your desktop:

☑ Ask you for confirmation

☑ Require the user to enter this password:

Password: ********

- To set up a persistent VNC server on Red Hat Enterprise Linux 4:
    - o Set a password using **vncpasswd**
    - o Edit **/etc/sysconfig/vncservers**
    - o Enable the service with **chkconfig vncserver on**
    - o Start the service with **service vncserver start**
    - o Edit **/home/username/.vnc/xstartup** if you want a more advanced session than just *twm* and an *xterm*

C. For Macintosh servers (and clients):

OSXvnc http://www.redstonesoftware.com/vnc.html is a robust, full-featured VNC server for Mac OS X that allows any VNC client to remotely view and/or control Mac OS X machine. OSXvnc is supported by Redstone Software

D. Most other operating systems (Solaris, HPUX, PalmOS etc) either come with VNC bundled, or have third-party VNC software that you can download

6.9.2 Install, configure and connect the VNC Viewer

VNC is truly platform-independent, so a VNC Viewer on any operating system can connect to a VNC Server on any other operating system. There are Viewers (and Servers) from a wide selection of sources (e.g. UltraVNC TightVNC or RealVNC) for most operating systems. There are also a wealth of Java viewers available so that any desktop can be viewed with any Java-capable browser (http://en.wikipedia.org/wiki/VNC lists many of the VNC Viewers sources).

- Install the VNC Viewer software and set it up for the appropriate speed connection

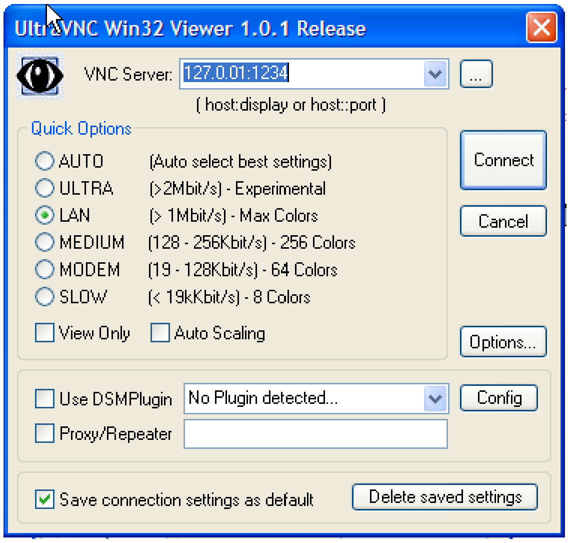
***Note**: To make VNC faster, when you set up the Viewer:*

- *Set encoding to ZRLE (if you have a fast enough CPU)*
- *Decrease color level (e.g. 64 bit)*
- *Disable the background transmission on the Server or use a plain wallpaper*

(Refer to http://doc.uvnc.com for detailed configuration instructions)

- To establish the VNC connection, first configure the VNC Viewer, entering the VNC Server IP address

A. When the Viewer computer is connected to the Console Server through an SSH tunnel (over the public Internet, or a dial-in connection, or private network connection), enter localhost (or 127.0.0.1) as the IP VNC Server IP address and the source port you entered when setting SSH tunneling/port forwarding (in Section 6.2.6) e.g. :1234



UltraVNC Win32 Viewer 1.0.1 Release

VNC Server: 127.0.01:1234

(host:display or host::port)

Quick Options

○ AUTO (Auto select best settings)
○ ULTRA (>2Mbit/s) - Experimental
● LAN (> 1Mbit/s) - Max Colors
○ MEDIUM (128 - 256Kbit/s) - 256 Colors
○ MODEM (19 - 128Kbit/s) - 64 Colors
○ SLOW (< 19kKbit/s) - 8 Colors

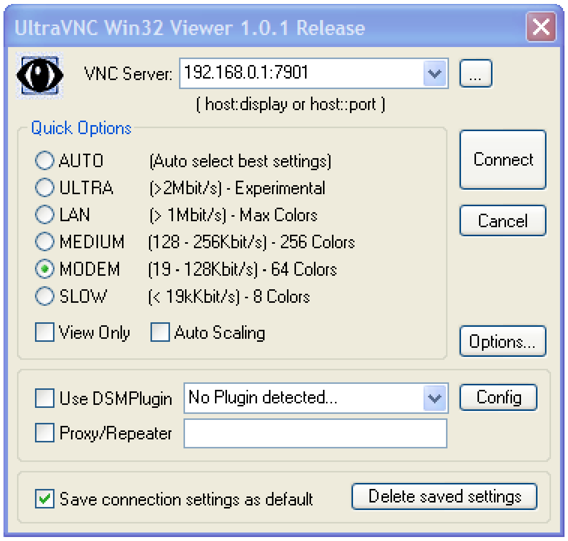
□ View Only □ Auto Scaling

□ Use DSMPlugin No Plugin detected...

□ Proxy/Repeater

☑ Save connection settings as default

B. When the Viewer computer is connected directly to the Console Server (either locally or remotely through a VPN or dial-in connection) and the VNC Host computer is serially connected to the Console Server, then 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 (i.e. 7901 to 7948, so all traffic directed to port 79xx on the Console Server is tunneled through to port 5900 on the PPP connection on serial Port xx). For example, for a Windows Viewer computer using UltraVNC connecting to a VNC Server which is attached to Port 1 on a Console Server, enter 192.168.0.1



UltraVNC Win32 Viewer 1.0.1 Release

VNC Server: 192.168.0.1:7901

(host:display or host::port)

Quick Options

- ○ AUTO (Auto select best settings)
- ○ ULTRA (>2Mbit/s) - Experimental
- ○ LAN (> 1Mbit/s) - Max Colors
- ○ MEDIUM (128 - 256Kbit/s) - 256 Colors
- ● MODEM (19 - 128Kbit/s) - 64 Colors
- ○ SLOW (< 19kKbit/s) - 8 Colors

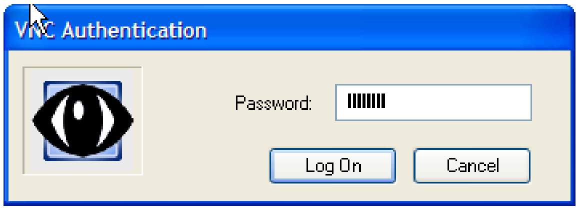
□ View Only □ Auto Scaling

□ Use DSMPlugin No Plugin detected...

□ Proxy/Repeater

☑ Save connection settings as default

- You can then establish the VNC connection by simply activating the VNC Viewer software on the Viewer computer and entering the password



VNC Authentication

Password: IIIIIIII

***Note**: For general 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
- ***Taking your desktop virtual with VNC**, Red Hat magazine*
  http://www.redhat.com/magazine/006apr05/features/vnc/ and http://www.redhat.com/magazine/007may05/features/vnc/
- ***Wikipedia** general background on VNC* http://en.wikipedia.org/wiki/VNC

6.10 SDT IP Connection to Hosts

Network (IP) protocols like RDP, VNC and HTTP can also be used to connect to host devices that are serially connected through their COM port to the Console Server. To do this you must:

- establish a PPP connection (Section 6.7.1) between the host and the gateway, then
- set up Secure Tunneling - Ports on the Console Server (Section 6.7.2), then
- 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 (Section 6.7.3)

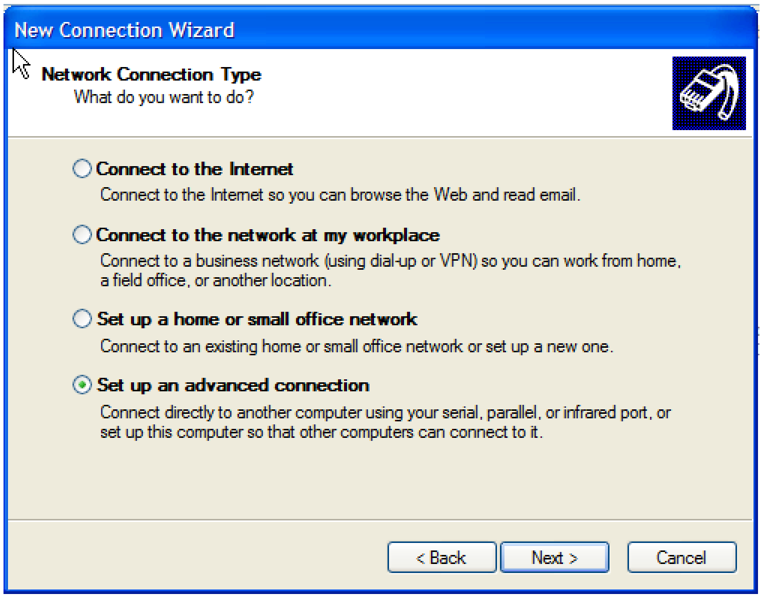
6.10.1 Establish a PPP connection between the host COM port and Console Server

(This step is only necessary for serially connected computers)
Firstly, physically connect the COM port on the host computer that is to be accessed to the serial port on the Console Server. Then:

A. For non-Windows computers (Linux, UNIX, Solaris etc), establish a PPP connection over the serial port. The online tutorial http://www.yolinux.com/TUTORIALS/LinuxTutorialPPP.html presents a selection of methods for establishing a PPP connection for Linux

B. For Windows XP and 2003 computers, follow the steps below to set up an advanced network connection between the Windows computer, through its COM port, to the Console Server. Both Windows 2003 and Windows XP Professional allow you to create a simple dial-in service which can be used for the Remote Desktop/VNC/HTTP/X connection to the Console Server:

- Open **Network Connections** in Control Panel and click the **New Connection Wizard**



New Connection Wizard

Network Connection Type
What do you want to do?

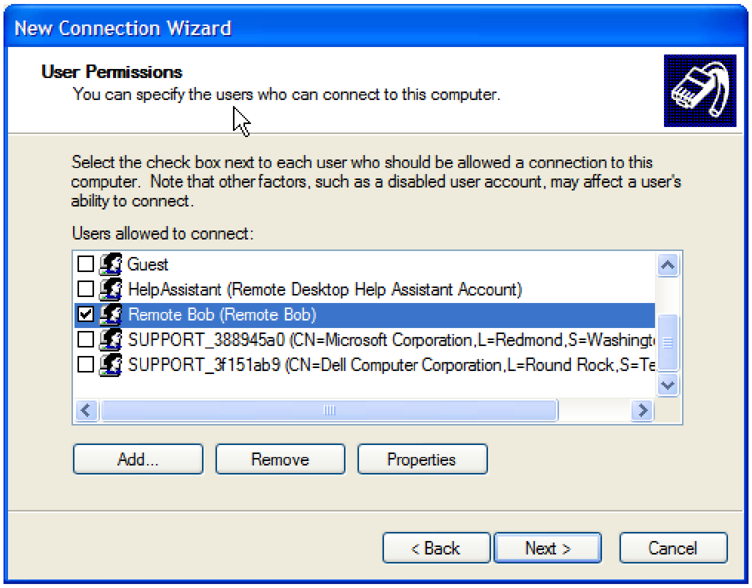
○ **Connect to the Internet**
Connect to the Internet so you can browse the Web and read email.

○ **Connect to the network at my workplace**
Connect to a business network (using dial-up or VPN) so you can work from home, a field office, or another location.

○ **Set up a home or small office network**
Connect to an existing home or small office network or set up a new one.

● **Set up an advanced connection**
Connect directly to another computer using your serial, parallel, or infrared port, or set up this computer so that other computers can connect to it.

- Select **Set up an advanced connection** and click **Next**
- On the **Advanced Connection Options** screen, select **Accept Incoming Connections** and click **Next**
- Select the **Connection Device** (i.e. the serial COM port on the Windows computer that you cabled through to the Console Server). By default, select **COM1**. The COM port on the Windows computer should be configured to its maximum baud rate. Click **Next**
- On the **Incoming VPN Connection Options** screen, select **Do not allow virtual private connections** and click **Next**



User Permissions

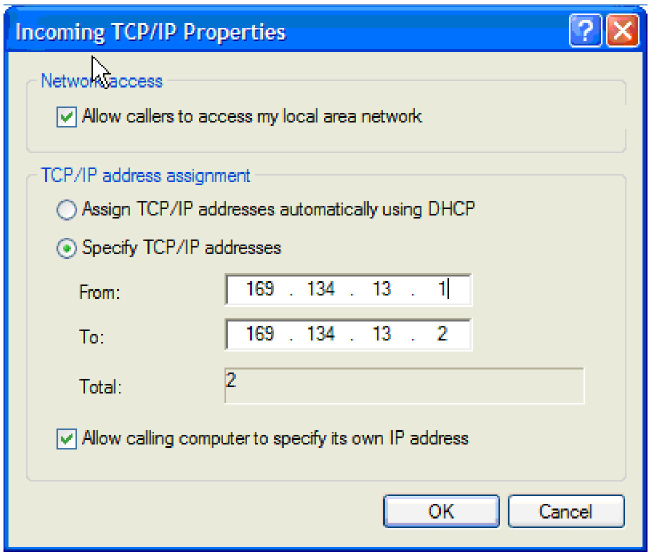
You can specify the users who can connect to this computer.

Select the check box next to each user who should be allowed a connection to this computer. Note that other factors, such as a disabled user account, may affect a user's ability to connect.

Users allowed to connect:

- □ Guest
- □ HelpAssistant (Remote Desktop Help Assistant Account)
- ☑ Remote Bob (Remote Bob)
- □ SUPPORT_388945a0 (CN=Microsoft Corporation,L=Redmond,S=Washingt
- □ SUPPORT_3f151ab9 (CN=Dell Computer Corporation,L=Round Rock,S=Te

- Specify which Users will be allowed to use this connection. This should be the same Users who were given Remote Desktop access privileges in the earlier step. Click **Next**
- On the **Network Connection** screen, select **TCP/IP** and click **Properties**



Network access

☑ Allow callers to access my local area network

TCP/IP address assignment

○ Assign TCP/IP addresses automatically using DHCP

● Specify TCP/IP addresses

From: 169 . 134 . 13 . 1

To: 169 . 134 . 13 . 2

Total: 2

☑ Allow calling computer to specify its own IP address

- Select **Specify TCP/IP addresses** on the **Incoming TCP/IP Properties** screen. Nominate a *From:* and a *To:* TCP/IP address and click **Next**

***Note**: You can choose any TCP/IP addresses as long as they are addresses which are not used anywhere else on your network. The **From:** address will be assigned to the Windows XP/2003 computer and the **To:** address will be used by the Console Server. For simplicity, use the IP address as shown in the illustration above:*

From: 169.134.13.1
To: 169.134.13.2

Alternately you can set the advanced connection and access on the Windows computer to use the Console Server defaults:

- *Specify 10.233.111.254 as the From: address*
- *Select **Allow calling computer to specify its own address***

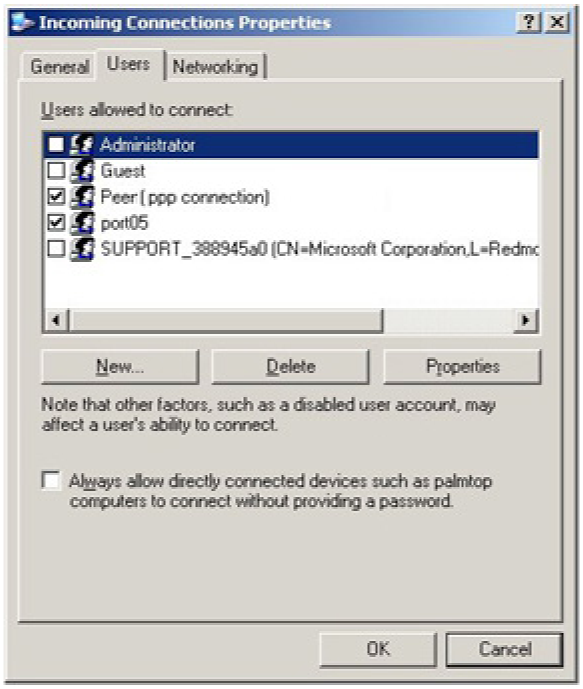
Also you could use the Console Server default username and password when you set up the new Remote Desktop User and give this User permission to use the advance connection to access the Windows computer:

- *The Console Server default **Username is portXX** where XX is the serial port number on the Console Server.*
- *The default **Password is portXX***

*So to use the defaults for an RDP connection to the serial port 2 on the Console Server, you would have set up a Windows user named **port02***

- When the PPP connection has been set up, a network icon will appear in the Windows task bar

***Note**: The above notes describe setting up an incoming connection for Windows XP. The steps are the same for Windows 2003, except that the setup screens present slightly differently:*



Users allowed to connect:

- □ Administrator
- □ Guest
- ☑ Peer (ppp connection)
- ☑ port05
- □ SUPPORT_388945a0 (CN=Microsoft Corporation,L=Redmc

Note that other factors, such as a disabled user account, may affect a user's ability to connect.

□ Always allow directly connected devices such as palmtop computers to connect without providing a password.

*Put a check in the box for **Always allow directly connected devices such as palmtop.....***

*Also, the option to **Set up an advanced connection** is not available in Windows 2003 if RRAS is configured. If RRAS has been configured, it is a simple task to enable the null modem connection for the dial-in configuration*.

C. For earlier version Windows computers, follow the steps in Section B, above. To get to the **Make New Connection** button:

- For Windows 2000, click **Start** and select **Settings**. At the **Dial-Up Networking Folder**, click **Network and Dial-up Connections** and click **Make New Connection**. Note: you first may need to set up a connection over the COM port using **Connect directly to another computer** before proceeding to **Set up an advanced connection**
- For Windows 98, you double-click **My Computer** on the Desktop, then open **Dial-Up Networking** and double-click

6.10.2 Set up SDT Serial Ports on Console Server

To set up *RDP (and VNC) forwarding* on the Console Server's Serial Port that is connected to the Windows computer COM port:

- Select the **Serial & Network: Serial Port** menu option and click **Edit** (for the particular Serial Port that is connected to the Windows computer COM port)
- On the SDT Settings menu, select **SDT Mode** (which will enable port forwarding and SSH tunneling) and enter a **Username** and **User Password**.

***Note**: When you enable SDT, this will override all other Configuration protocols on that port*

Note**: If you leave the **Username** and **User Password** fields blank, they default to **portXX** and **portXX** where XX is the serial port number. So the default username and password for Secure RDP over Port 2 is **port02

- Ensure the Console Server **Common Settings** (Baud Rate, Flow Control) are the same as were set up on the Windows computer COM port and click **Apply**
- RDP and VNC forwarding over serial ports is enabled on a Port basis. You can add Users who can have access to these ports (or reconfigure User profiles) by selecting **Serial & Network :User & Groups** menu tag - as described earlier in *Chapter 4 Configuring Serial Ports*

6.10.3 Set up SDT Connector to SSH port forward over the Console Server Serial Port

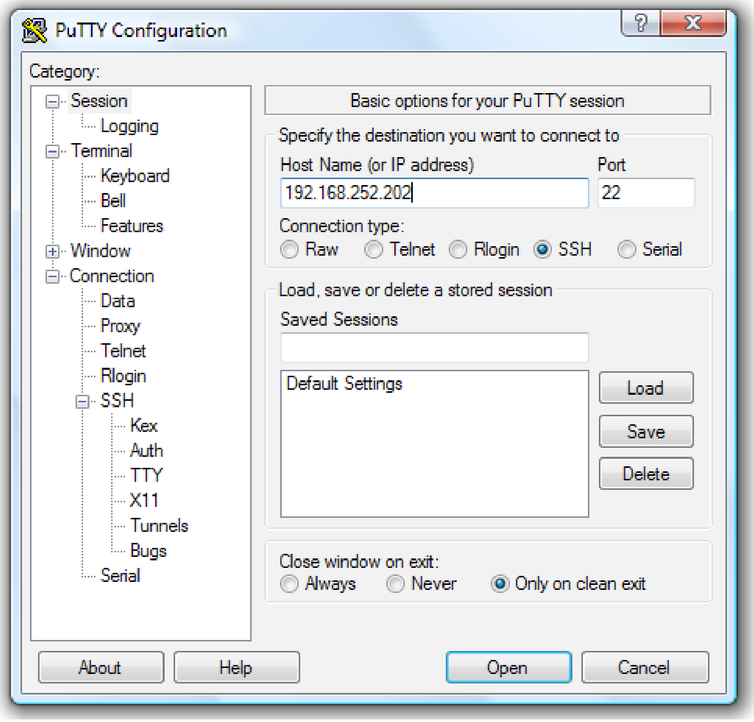
In the *SDT Connector* software running on your remote computer, specify the gateway IP address of your Console Server and a username/password for a user you have setup on the Console Server that has access to the desired port.
Next, add a New SDT Host. In the Host address you need to put portxx where xx = the port to which you are connecting. Example, for port 3 you would have a Host Address of: port03 and then select the RDP Service check box.

6.11 SSH Tunneling using other SSH clients (e.g. PuTTY)

As covered in the previous sections of this chapter we recommend you use the *SDT Connector* client software that is supplied with the Console Server. However there's also a wide selection of commercial and free SSH client programs that can also provide the secure SSH connections to the Console Servers and secure tunnels to connected devices:

- PuTTY is a complete (though not very user friendly:) freeware implementation of SSH for Win32 and UNIX platforms
- SSHTerm is a useful open source SSH communications package
- SSH Tectia is leading end-to-end commercial communications security solution for the enterprise
- Reflection for Secure IT (formerly F-Secure SSH) is another good commercial SSH-based security solution

By way of example the steps below show the establishment of an SSH tunneled connection to a network connected device using the PuTTY client software.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.252.202

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

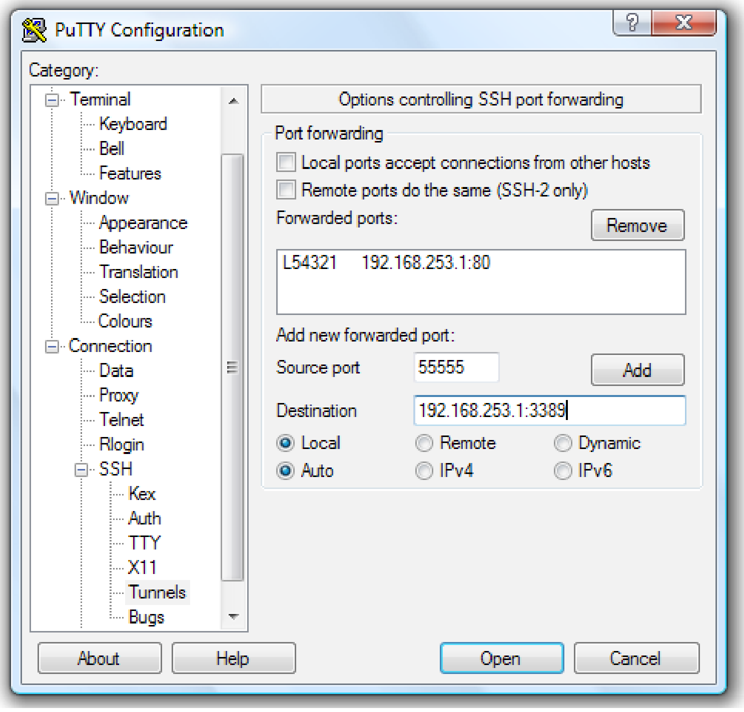
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

- In the **Session** menu enter the IP address of the Console Server in the **Host Name or IP address** field
    - o 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
    - o For Internet (or local*/VPN connections) connections this will be the public IP address of the Console Server*
- Select the **SSH Protocol**, and the **Port** will be set as 22
- Go to the **SSH: Tunnels** menu and in *Add new forwarded* port enter any high unused port number for the **Source port** e.g. 54321
- Set the **Destination**: IP details
    - o If your destination device is network connected to the Console Server and you are connecting using RDP, set the Destination as *<Managed Device IP address/DNS Name>:3389* e.g. if when setting up the Managed Device as *Network Host* on the Console Server you specified its IP address to be 192.168.253.1 (or its DNS Name was *accounts.myco.intranet.com*) then specify the Destination as *192.168.523.1:3389* (or *accounts.myco.intranet.com:3389*). Only devices which 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.



□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH-2 only)

Forwarded ports:

```
L54321     192.168.253.1:80
```

Source port: 55555

Destination: 192.168.253.1:3389

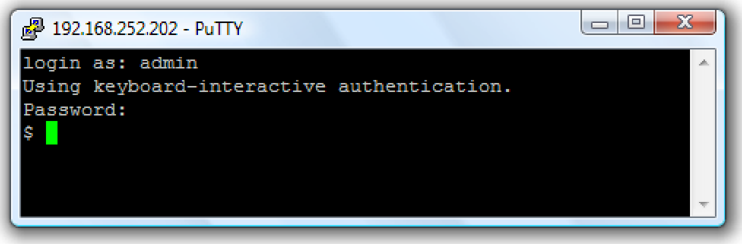
● Local ○ Remote ○ Dynamic

● Auto ○ IPv4 ○ IPv6

  - If your destination computer is serially connected to the Console Server, set the Destination as <port label>:3389 e.g. if the Label you specified on the serial port on the Console Server is win2k3, then specify the remote host as win2k3:3389 . Alternative you can set the Destination as *portXX:3389* where XX is the SDT enabled serial port number e.g. if port 4 is on the Console Server is to carry the RDP traffic then specify *port04:3389*

***Note**: http://www.jfitz.com/tips/putty_config.html has useful examples on configuring PuTTY for SSH tunneling*

- Select **Local** and click the **Add** button
- Click **Open** to SSH connect the Client PC to the Console Server. You will now be prompted for the Username/Password for the Console Server user



```
login as: admin
Using keyboard-interactive authentication.
Password:
$ 
```

- If you are connecting as a User in the “users” group then you can only SSH tunnel to Hosts and Serial Ports where you have specific access permissions
- If you are connecting as an Administrator (in the “admin” group) then you can connect to any configured Host or Serial Ports (whic*h 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 as was used for RDP) in the Destination IP address.

To set up the secure SSH tunnel from the Client (Viewer) PC to the Console Server for VNC follow the steps above, however when configuring the VNC port redirection specify port 5900 in the Destination IP address.

Note*: 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.*
However, once connected, all subsequent VNC traffic is unencrypted. So a malicious user could snoop your VNC session. Also there are VNC scanning programs available, which 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. Also 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 which you're opening on your Console Server the SDT port 22.
So sometimes 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 7: Alerts, Automated Response and Logging

This chapter describes the automated response, alert generation and logging features of the Console Server.

The new Auto-Response facility (in firmware V3.5.1 and later) extends on the basic Alert facility available in earlier firmware revisions. With the new facility 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 the triggers. To configure, you:

- Set general parameters then select and configure the *Check Conditions* i.e. the conditions that will trigger the response (*Section 7.1*), then
- Specify the *Trigger Actions* i.e. sequence of actions initiated in the event of the trigger condition, then specify the *Resolve Actions* i.e. actions performed when trigger conditions have been resolved (*Section 7.2*)

The Console Servers can also be configured selectively to maintain log records of all access and communications with the Console Server and with the attached serial devices, all system activity and a history of the status of any attached environmental monitors, UPS and PDU devices. The Console Servers can also log access and communications with network attached hosts.

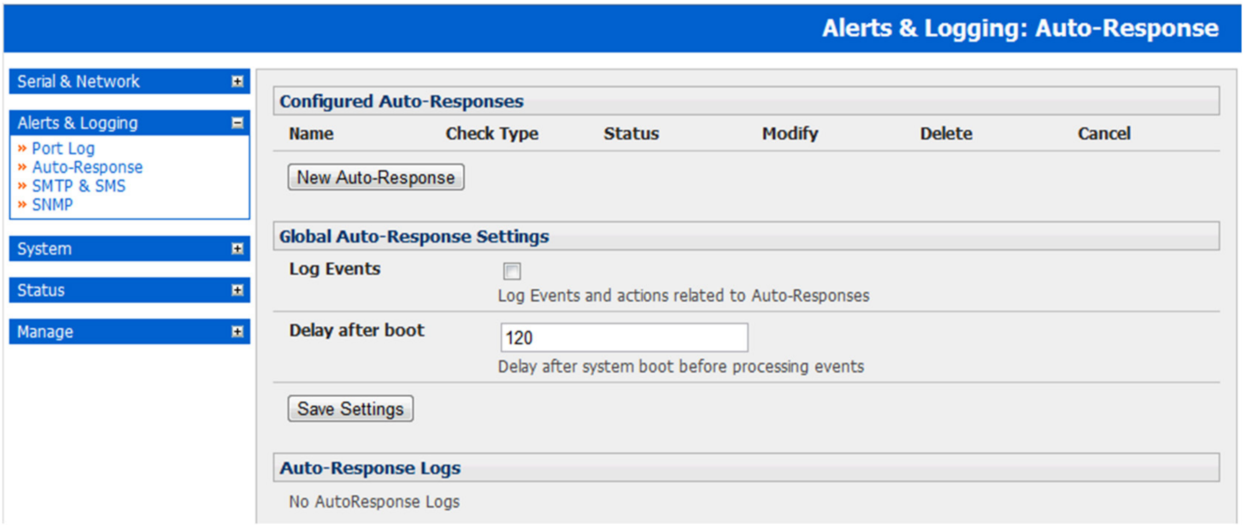
- If port logs are to be maintained on a remote server, then the access path to this location needs to be configured (*Section 7.3*)
- Then you need to activate and set the desired levels of logging for each serial (*Section 7.4*) and/or network port (*Section 7.5*) and/or power and environment devices (refer to *Chapter 8*)

7.1 Set Up Auto-Response and Configure Check Conditions

With the Auto-Response facility, a sequence of Trigger Actions is initiated in the event of a specified trigger condition (Check Condition). Subsequent Resolve Actions can also be performed when the trigger condition has been resolved.

To configure, first set the general parameters that will be applied to all Auto-Responses:

- Check Log Events on Alerts & Logging: Auto-Response to enable logging all Auto-Response activities
- Check Delay after Boot to set any general delay to be applied after console server system boot, before processing events



Alerts & Logging: Auto-Response

Configured Auto-Responses

Name	Check Type	Status	Modify	Delete	Cancel

Global Auto-Response Settings

Log Events □ Log Events and actions related to Auto-Responses

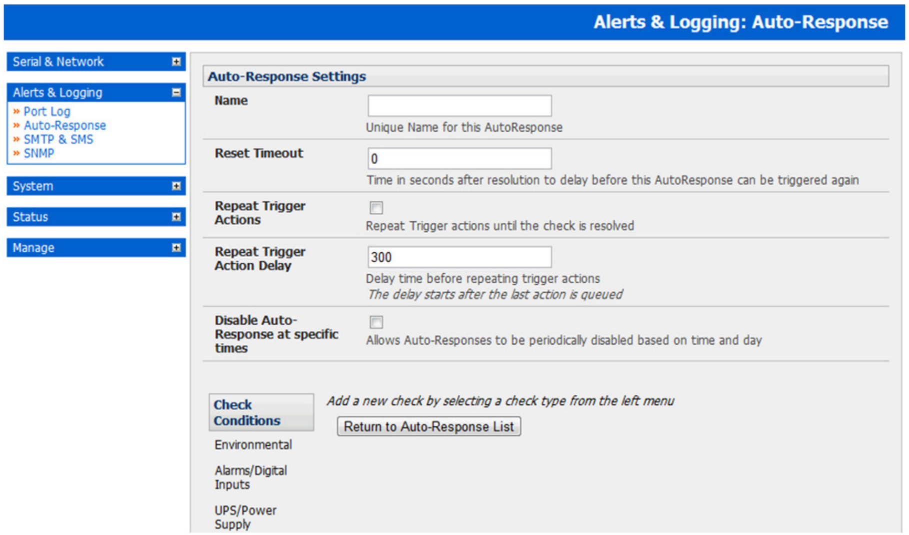
Delay after boot 120 Delay after system boot before processing events

Auto-Response Logs

No AutoResponse Logs

To configure a new Auto-Response:

- Select New Auto-Response in the Configured Auto-Response field. You will be presented with a new Auto-Response Settings menu
- 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 continue to repeat trigger action sequences 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



Auto-Response Settings

Name [ ]
Unique Name for this AutoResponse

Reset Timeout [0]
Time in seconds after resolution to delay before this AutoResponse can be triggered again

Repeat Trigger Actions □
Repeat Trigger actions until the check is resolved

Repeat Trigger Action Delay [300]
Delay time before repeating trigger actions
The delay starts after the last action is queued

Disable Auto-Response at specific times □
Allows Auto-Responses to be periodically disabled based on time and day

Check Conditions
Environmental
Alarms/Digital Inputs
UPS/Power Supply

Add a new check by selecting a check type from the left menu

Return to Auto-Response List

- Check Disable Auto-Response at specific times and you will be able to periodically disable Auto-Responses between specified times of day

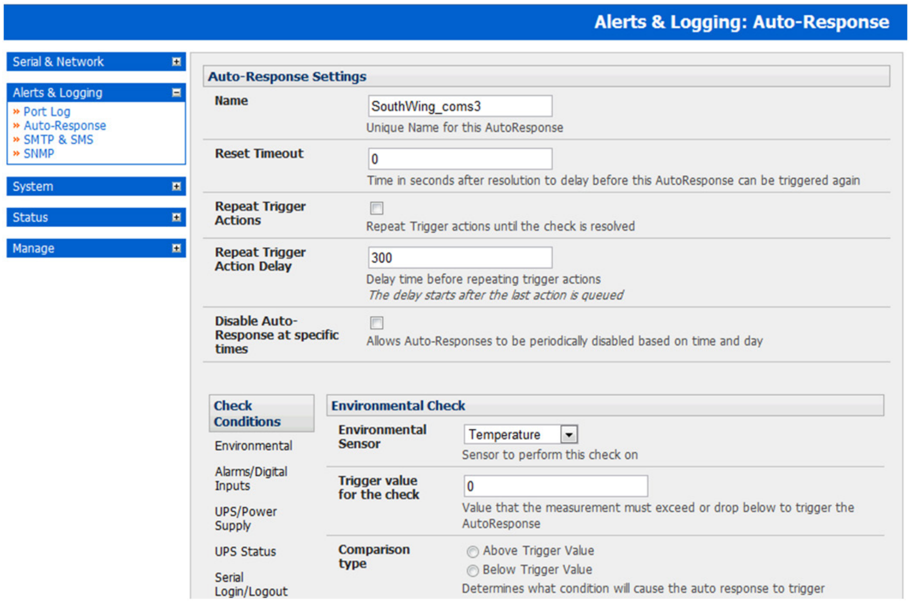
To configure the condition that will trigger the Auto-Response:

- Click on the Check Condition type (e.g. Environmental, UPS Status or ICMP ping) to be configured as the trigger for this new Auto-Response in the Auto-Response Settings menu

7.1.1 Environmental Check

To configure Humidity or Temperature levels as the trigger event:

- Click on the Environmental as the Check Condition



Auto-Response Settings

Name [SouthWing_coms3]
Unique Name for this AutoResponse

Reset Timeout [0]
Time in seconds after resolution to delay before this AutoResponse can be triggered again

Repeat Trigger Actions □
Repeat Trigger actions until the check is resolved

Repeat Trigger Action Delay [300]
Delay time before repeating trigger actions
The delay starts after the last action is queued

Disable Auto-Response at specific times □
Allows Auto-Responses to be periodically disabled based on time and day

Check Conditions
Environmental
Alarms/Digital Inputs
UPS/Power Supply
UPS Status
Serial Login/Logout

Environmental Check

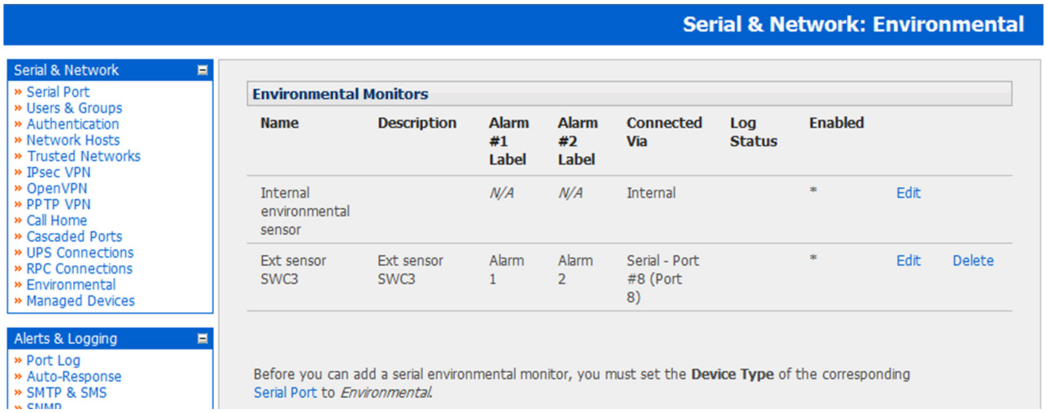
Environmental Sensor [Temperature ▾]
Sensor to perform this check on

Trigger value for the check [0]
Value that the measurement must exceed or drop below to trigger the AutoResponse

Comparison type
○ Above Trigger Value
○ Below Trigger Value
Determines what condition will cause the auto response to trigger

- In the Environmental Check menu, select the specific Environmental Sensor to be checked for the trigger
- Specify the Trigger value (in °C / °F for Temp 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 (e.g. 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 temp 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 Temp and/ or Humidity sensors on your attached EMD*



Environmental Monitors

Name	Description	Alarm #1 Label	Alarm #2 Label	Connected Via	Log Status	Enabled		
Internal environmental sensor		*N/A*	*N/A*	Internal		*	Edit	
Ext sensor SWC3	Ext sensor SWC3	Alarm 1	Alarm 2	Serial - Port #8 (Port 8)		*	Edit	Delete

Before you can add a serial environmental monitor, you must set the **Device Type** of the corresponding Serial Port to *Environmental*.

7.1.2 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 ACM5000*

7.1.3 UPS / Power Supply

To use the properties of any attached UPS as the trigger event:

- Click on UPS / Power Supply as the Check Condition
- Select UPS Power Device Property (Input Voltage, Battery Charge %, Load %, Input Frequency Hz or Temperature in °C) that will checked for the trigger
- Specify the Trigger value that the check measurement must exceed or drop below to trigger the Auto-Response
- 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 (e.g. if an Auto-Response was set up with a trigger event of a battery charge below 20% with a Hysteresis of 5 then the trigger condition would not be seen as having been resolved till the battery charge was above 25%)
- Check Save Auto-Response

***Note**: Before configuring UPS checks in Auto-Response you first must configure the attached UPS*

7.1.4 UPS Status

To use the alert state of any attached UPS as the Auto-Response trigger event:

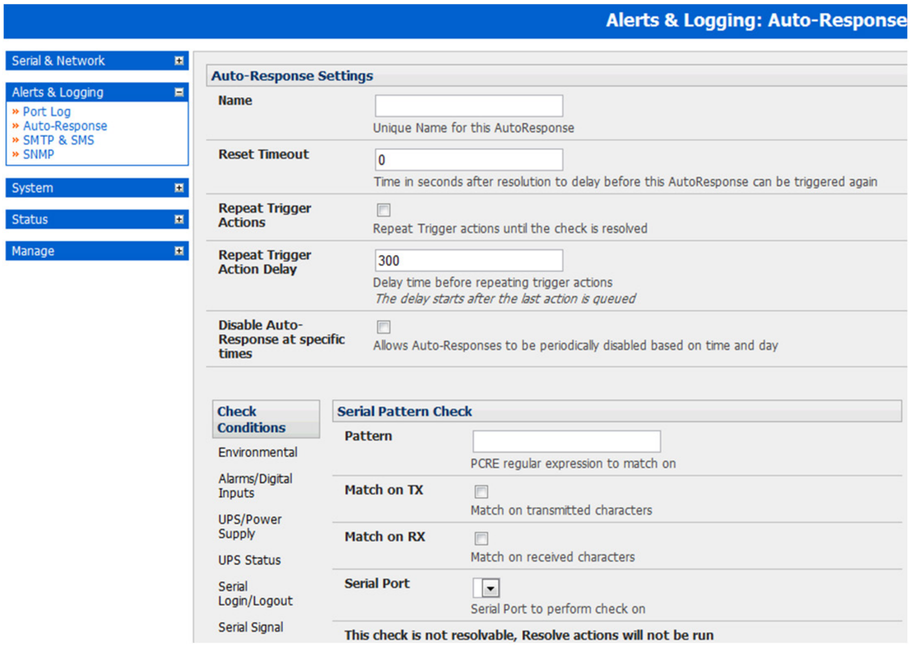
- Click on UPS Status as the Check Condition
- Select the reported UPS State to trigger the Auto-Response (either On Battery or Low Battery). The Auto-Response will resolve when the UPS state returns to the “Online” state
- Select which connected UPS Device to monitor and check Save Auto-Response

***Note**: Before configuring UPS state checks in Auto-Response you first must configure the attached UPS*

7.1.5 Serial Login/Logout

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. Then 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 and specify Serial Port to perform check on, and/or
- Click on Serial Signal as the Check Condition. Then in the Serial Signal Check menu select the Signal (CTS, DCD, DSR) to trigger on, the Trigger condition (either on serial signal change, or check level) and specify Serial Port to perform check on, and/or
- Click on Serial Pattern as the Check Condition. Then 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
- Check Save Auto-Response



Auto-Response Settings

Name	Unique Name for this AutoResponse
Reset Timeout	0 Time in seconds after resolution to delay before this AutoResponse can be triggered again
Repeat Trigger Actions	□ Repeat Trigger actions until the check is resolved
Repeat Trigger Action Delay	300 Delay time before repeating trigger actions *The delay starts after the last action is queued*
Disable Auto-Response at specific times	□ Allows Auto-Responses to be periodically disabled based on time and day

Serial Pattern Check

Pattern	PCRE regular expression to match on
Match on TX	□ Match on transmitted characters
Match on RX	□ Match on received characters
Serial Port	Serial Port to perform check on

This check is not resolvable, Resolve actions will not be run

***Note**: Before configuring serial port checks in Auto-Response you first must configure the serial port in Console Server mode. Also, most serial port checks are not resolvable so resolve actions will not be run.*

7.1.6 ICMP Ping

To use a ping result as the Auto-Response trigger event:

- Click on ICMP Ping as the Check Condition
- Specify which Address to Ping (i.e. IP address or DNS name to send ICMP Ping to) and which Interface to send ICMP Ping from (e.g. Management LAN or Wireless network)
- Set the Check Frequency (i.e. the time in seconds between checks) and the Number of ICMP Ping packets to send
- Check Save Auto-Response

7.1.7 Cellular Data

This check monitors the aggregate data traffic inbound and outbound through the cellular modem as an Auto-Response trigger event.

- Click on Cellular Data as the Check Condition

***Note**: Before configuring cellular data checks in Auto-Response the internal or external USB cellular modem must be configured and detected by the console server.*

7.1.8 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 e.g. /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
```

- Enter the Script Executable file name (e.g. /etc/config/test.sh)
- Set the Check Frequency (i.e. the time in seconds between re-running the script) and the Script Timeout (i.e. 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 (e.g. with a web page html check script, these Arguments might specify the web page address/DNS and user logins)
- Check Save Auto-Response

7.1.9 SMS Command

An incoming SMS command from a nominated caller can trigger an Auto-Response:

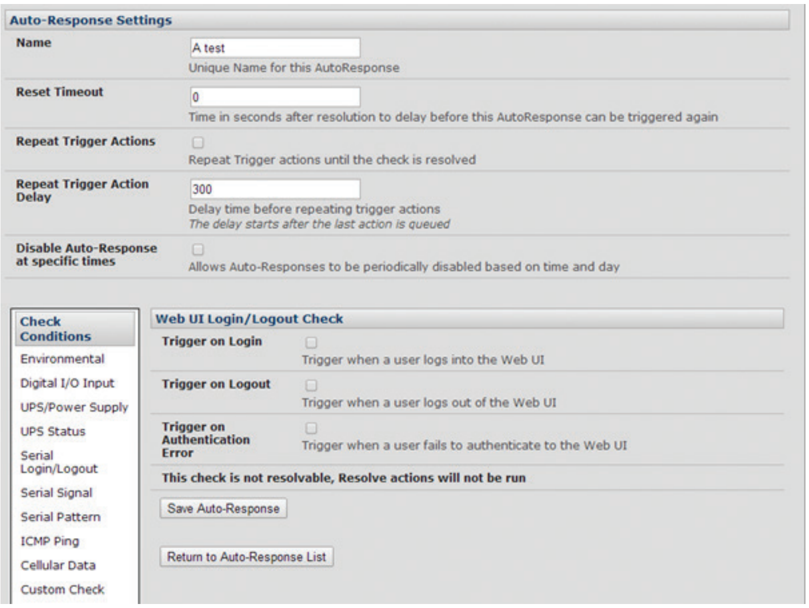
- Click on SMS Command as the Check Condition
- Specify which Phone Number (in international format) of the phone sending the SMS message
- Set the Incoming Message Pattern (PCRE regular expression) to match to create trigger event

***Note**: The SMS command trigger condition can only be set if there is an external USB cellular modem detected*

7.1.10 Log In/Out Check

To configure Web Log In/Out as the trigger event:

- Click on the **Web UI Authentication** as the **Check Condition**



Auto-Response Settings

Name: A test
Unique Name for this AutoResponse

Reset Timeout: 0
Time in seconds after resolution to delay before this AutoResponse can be triggered again

Repeat Trigger Actions: □
Repeat Trigger actions until the check is resolved

Repeat Trigger Action Delay: 300
Delay time before repeating trigger actions
The delay starts after the last action is queued

Disable Auto-Response at specific times: □
Allows Auto-Responses to be periodically disabled based on time and day

Web UI Login/Logout Check

Trigger on Login: □
Trigger when a user logs into the Web UI

Trigger on Logout: □
Trigger when a user logs out of the Web UI

Trigger on Authentication Error: □
Trigger when a user fails to authenticate to the Web UI

This check is not resolvable, Resolve actions will not be run

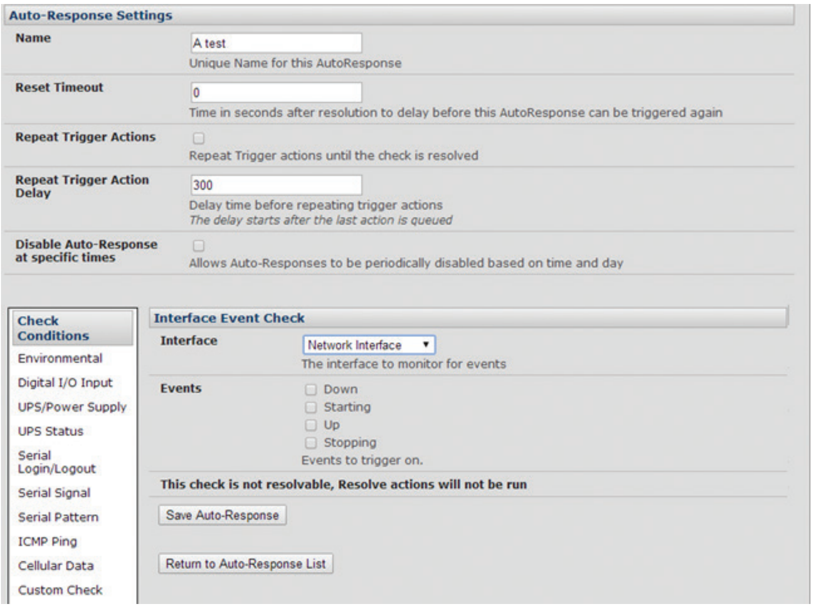
- Check **Trigger on Login (Logout)** to trigger when a user logs into (or out of) the Web UI
- Check **Trigger on Authentication Error** to trigger when a user fails to authenticate to the Web UI

Note: *This check is not resolvable and Resolve actions will not run*

7.1.11 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):

- Click on **Network Interface** as the **Check Condition**



Auto-Response Settings

Name: A test
Unique Name for this AutoResponse

Reset Timeout: 0
Time in seconds after resolution to delay before this AutoResponse can be triggered again

Repeat Trigger Actions: □
Repeat Trigger actions until the check is resolved

Repeat Trigger Action Delay: 300
Delay time before repeating trigger actions
The delay starts after the last action is queued

Disable Auto-Response at specific times: □
Allows Auto-Responses to be periodically disabled based on time and day

Interface Event Check

Interface: Network Interface
The interface to monitor for events

Events: □ Down □ Starting □ Up □ Stopping
Events to trigger on.

This check is not resolvable, Resolve actions will not be run

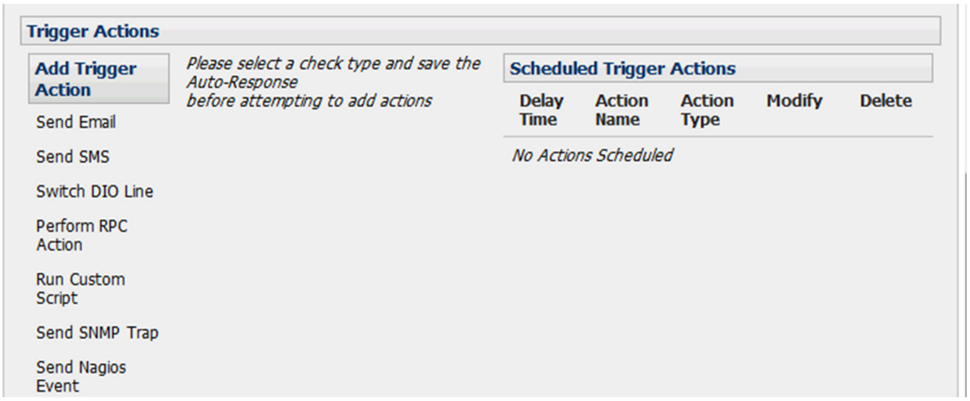
- Select the **Interface** (Ethernet /Failover OOB Interface or Modem or VPN) to monitor
- Check what type of network interface **Event** to trigger on (interface Down, Starting, Up or Stopping)

Note: *This check is not resolvable and Resolve actions will not run*

7.2 Trigger and Resolve Actions

To configure the sequence of actions taken in the event of the trigger condition:

- For a nominated Auto-Response - with a defined Check Condition - click on Add Trigger Action (e.g. Send Email or Run Custom Script) to select the action type to be taken. Then configure the selected action (as detailed in the following sections)
- 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. So 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



Trigger Actions

Add Trigger Action

Send Email

Send SMS

Switch DIO Line

Perform RPC Action

Run Custom Script

Send SNMP Trap

Send Nagios Event

Please select a check type and save the Auto-Response before attempting to add actions

Scheduled Trigger Actions

Delay Time	Action Name	Action Type	Modify	Delete
No Actions Scheduled				

***Note**: A message text can be sent with Email, SMS and Nagios actions. This configurable message can include selected values:*

$AR_TRIGGER_VAL = the trigger value for the check e.g. for UPS Status, it could be onbatt or battlow
$AR_VAL = the value returned by the check e.g. for ups status, it could be online/onbatt/battlow
$AR_CHECK_DEV = the device name of the device being checked e.g. 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

7.2.1 Send Email on Trigger

- Click on Send Email as the Add Trigger Action. Enter a unique Action Name and set the Action Delay Time
- Specify the Recipient Email Address to send this email to and the Subject of the email. For multiple recipients you can enter comma separated addresses
- Edit the Email Text message to send and 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 (e.g. for T-Mobile it is phonenumber @tmomail.net)*

7.2.2 Send SMS on Trigger

- Click on Send SMS as the Add Trigger Action. Enter a unique Action Name and set the Action Delay Time
- Specify the Phone number that the SMS will be sent to in international format (without the +)
- Edit the Message Text to send and click Save New Action

***Note**: The SMS alert can only be sent if there is an internal or external USB cellular modem attached. However an SMS alert can also be sent via a SMTP SMS gateway as described above.*

7.2.3 Perform RPC Action on Trigger

- Click on Perform RPC Action as the Add Trigger Action. Enter a unique Action Name and set the Action Delay Time
- Select a power Outlet and specify the Action to be performed (power On, OFF or Cycle)
- Click Save New Action

7.2.4 Run Custom Script on Trigger

- Click on Run Custom Script as the Add Trigger Action. Enter a unique Action Name and set the Action Delay Time
- Create a script file to execute when this action is triggered and enter the Script Executable file name e.g. /etc/config/action.sh
- Set the Script Timeout (i.e. the maximum run-time for the script). Leave as 0 for unlimited.
- Enter any Arguments that are to be passed to the script and click Save New Action

7.2.5 Send SNMP Trap on Trigger

- Click on Send SNMP Trap as the Add Trigger Action. Enter a unique Action Name and set the Action Delay Time

Note*: The SNMP Trap actions are valid for Serial, Environmental, UPS and Cellular data triggers only*

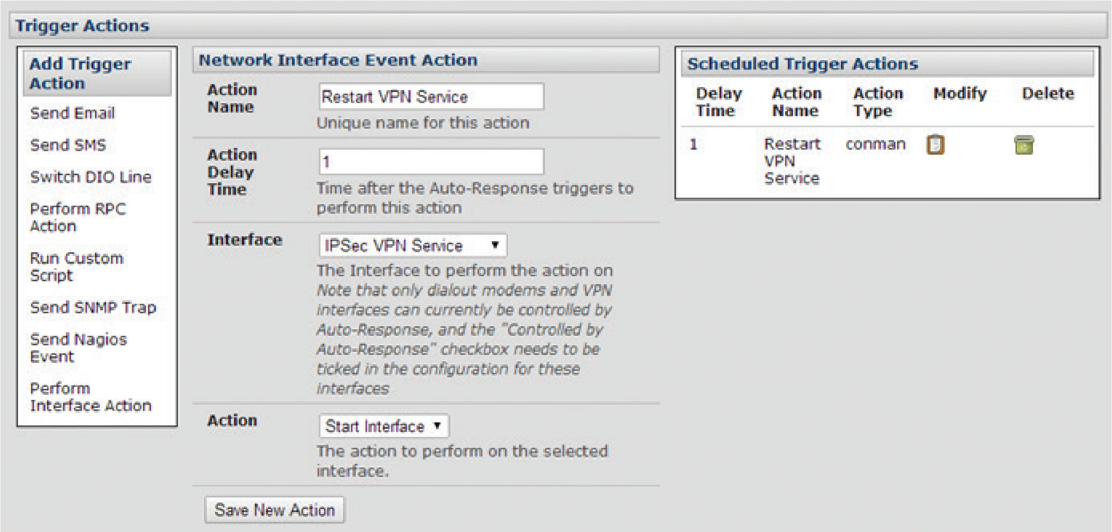
7.2.6 Send Nagios Event on Trigger

- Click on Send Nagios Event as the Add Trigger Action. Enter a unique Action Name and 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*

7.2.7 Perform Interface Action

- Click on **Perform Interface Action** as the **Add Trigger Action**. Enter a unique **Action Name** and set the **Action Delay Time**
- **Select the Interface** (Modem or VPN service) and the **Action** (Start or Stop Interface) to be taken



Network Interface Event Action

Action Name: Restart VPN Service
Unique name for this action

Action Delay Time: 1
Time after the Auto-Response triggers to perform this action

Interface: IPSec VPN Service
The Interface to perform the action on
Note that only dialout modems and VPN interfaces can currently be controlled by Auto-Response, and the "Controlled by Auto-Response" checkbox needs to be ticked in the configuration for these interfaces

Action: Start Interface
The action to perform on the selected interface.

Scheduled Trigger Actions

Delay Time	Action Name	Action Type	Modify	Delete
1	Restart VPN Service	conman	📋	🗑

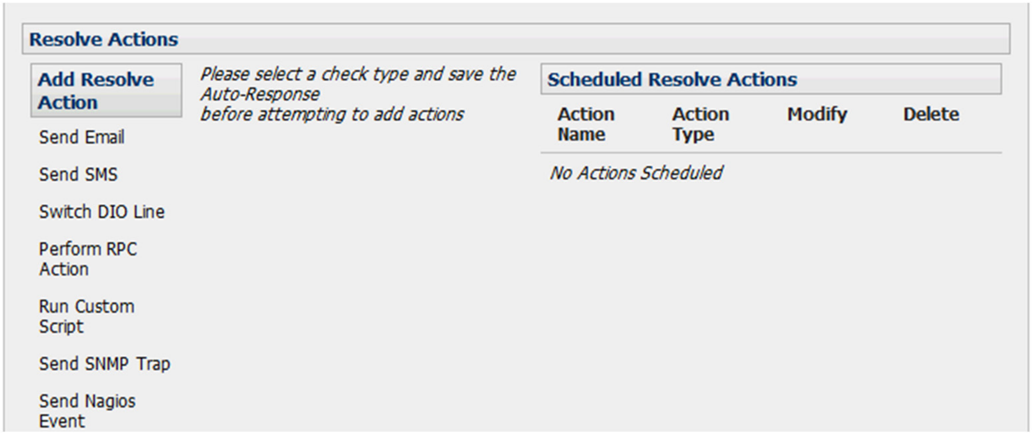
Note: *If any IPsec service or OpenVPN tunnel is to be controlled by the Network Interface Event Action, you will need to have checked the* Control by Auto-Response *box when configuring that service (if selected, the default state for the VPN tunnel / service will be Down).*

7.2.8 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 on Add Resolve Action (e.g. Send Email or Run Custom Script) to select the action type to be taken

***Note**: Resolve Actions are configured exactly the same as Trigger Actions except the designated Resolve Actions are all executed on resolution of the trigger condition and there are no Action Delay Times set*



Resolve Actions

Please select a check type and save the Auto-Response before attempting to add actions

Scheduled Resolve Actions

Action Name	Action Type	Modify	Delete
No Actions Scheduled			

Configure SMTP, SMS, SNMP and/or 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.

7.2.9 Send Email alerts on Resolution

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:

- Select Alerts & Logging: SMTP & SMS
- In the SMTP Server field enter the IP address of the outgoing mail Server
- If this mail server uses a Secure Connection, specify its type. You may also specify the IP port to use for SMTP. The default SMTP Port is 25.
- You may enter a Sender email address which will appear as the “from” address in all email notifications sent from this console server. 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
- You may also enter a Username and Password if the SMTP server requires authentication
- You can also specify the specific Subject Line that will be sent with the email
- Click Apply to activate SMTP

7.2.10 Send SMS alerts on Resolution

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 who provide email to SMS forwarding to phones on many 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.

SMS via Email Gateway

To use SMTP SMS, the Administrator must configure a valid SMTP server for sending the email:

- In the SMTP Settings field in the Alerts & Logging: SMTP & SMS menu select SMS Gateway. An SMS via Email Gateway field will appear
- Enter the IP address of the outgoing mail Server SMS gateway
- Select a Secure Connection (if applicable) and specify the SMTP port to be used (if other than the default port 25)
- You may also enter a Sender email address which will appear as the “from” address in all email notifications sent from this console server. Some SMS gateway service providers only forward email to SMS when the email has been received from authorized senders. So you may need to assign a specific authorized email address for the console server
- You may also enter a Username and Password as some SMS gateway service providers use SMTP servers which require authentication
- Similarly you can specify the specific Subject Line that will be sent with the email. 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). However some SMS gateway service providers require blank subjects or require specific authentication headers to be included in the subject line
- Click Apply Settings to activate SMS-SMTP connection.

SMS via Cellular Modem

To use an attached or internal cellular modem for SMS the Administrator must enable SMS:

- Select Cellular Modem In the SMS Settings field
- 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 Centre (only if advised by your carrier or Support)
- Click Apply Settings to activate SMS-SMTP connection

7.2.11 Send SNMP Trap alerts on Resolution

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:

- Select Alerts & Logging: SNMP
- Select Primary SNMP Manager tab. The Primary 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 then, at a minimum, 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 be configured to provide status information on demand using snmpd. This SNMP agent is configured using the SNMP Service Detail on Alerts & Logging: SNMP - as described in Chapter 15.*

- Select the Manager Protocol. SNMP is generally a UDP-based protocol, though it infrequently uses TCP instead.
- Enter the host address of the SNMP Network Manager into the Manager Address field.
- Enter the TCP/IP port number into the Manager Trap Port field (default =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.

- Configure SNMP v3 if required. 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.
  - o Enter the Engine ID for the user sending messages as a hex number (e.g. 0x8000000001020304).
  - o Specify the Security Level. The level of security has to be compatible with the settings of the remote SNMP Network Manager.

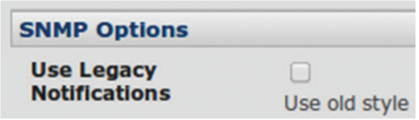
noAuthNoPriv	*No authentication or encryption.*
authNoPriv	*Authentication only. An authentication protocol (SHA or MD5) and password will be required.*
authPriv	*Uses both authentication and encryption. This is the highest level of security and requires an encryption protocol (DES or AES) and password in addition to the authentication protocol and password.*

  - o 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.
  - o An Authentication Protocol (SHA or MD5) and Authentication Password must be given for a Security Level of either authNoPriv or authPriv. The password must contain at least 8 characters to be valid.
  - o A Privacy Protocol (DES or AES) must be specified for the authPriv level of security to be used as the encryption algorithm. AES is recommended for stronger security. A password of at least 8 characters must be provided for encryption to work.
- Click Apply

***Note**: Console Servers with V3.0 firmware (and later) embed the net-snmpd daemon which can accept SNMP requests from remote SNMP management servers and provides information on serial port and device status (refer Chapter 15.5 for more details).*

Console servers with firmware earlier than V3.3 could only configure a Primary SNMP server from the Management Console. Refer Chapter 15.5 for details on configuring the snmptrap daemon to send traps/notifications to multiple remote SNMP servers.

***Note:** For firmware versions 3.10.2 and above, a 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 the SNMP menu to **Use Legacy Notifications** for the SNMP traps. In setting this option, the console server will send SNMP traps that are compatible with those sent from older firmware versions before new MIBS were added. This ensures that the firmware upgrade will not upset the existing SNMP management settings already in place.*



SNMP Options

Use Legacy Notifications □ Use old style

When upgrading from an old firmware version that does not support newer SNMP MIBs/traps (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, then the console server will be configured to use the new SNMP traps after the upgrade. Note: this will not have any effect until the SNMP service is turned on and an SNMP manager is configured.
- When starting up using the new firmware after a config erase, the console server will be configured to use the new SNMP traps.
- When upgrading from a firmware version that supports the new traps to a newer version that supports the new traps, the 'use legacy traps' setting should be kept the same – no checking SNMP service/manager configuration is needed.

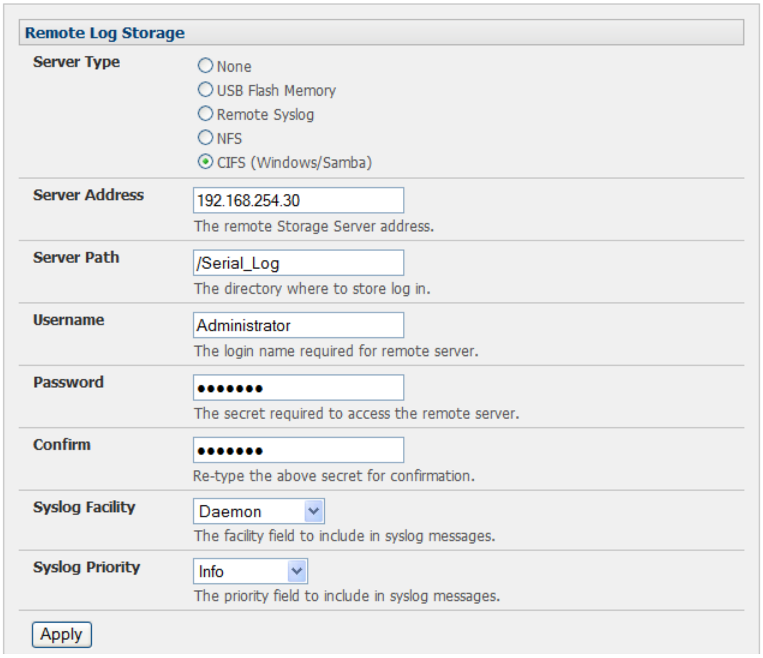
7.2.12 Send Nagios Event alerts on Resolution

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 (refer to Chapter 10).

7.3 Remote Log Storage

Before activating Serial or Network Port Logging on any port or UPS logging, you must specify where those logs are to be saved:

- Select the **Alerts & Logging: Port Log** menu option and specify the **Server Type** to be used, and the details to enable log server access



Remote Log Storage

Server Type
○ None
○ USB Flash Memory
○ Remote Syslog
○ NFS
● CIFS (Windows/Samba)

Server Address 192.168.254.30
The remote Storage Server address.

Server Path /Serial_Log
The directory where to store log in.

Username Administrator
The login name required for remote server.

Password •••••••
The secret required to access the remote server.

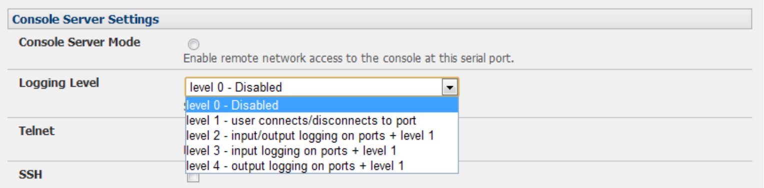
Confirm •••••••
Re-type the above secret for confirmation.

Syslog Facility Daemon
The facility field to include in syslog messages.

Syslog Priority Info
The priority field to include in syslog messages.

7.4 Serial Port Logging

In Console Server mode, activity logs can be maintained of all serial port activity. These records are stored on an off-server, or in the Console Server flash memory. To specify which serial ports are to have activities recorded and to what level data is to be logged:



Console Server Settings

Console Server Mode ○
Enable remote network access to the console at this serial port.

Logging Level level 0 - Disabled
level 0 - Disabled
level 1 - user connects/disconnects to port
level 2 - input/output logging on ports + level 1
level 3 - input logging on ports + level 1
level 4 - output logging on ports + level 1

Telnet

SSH □

- Select **Serial & Network: Serial Port** and **Edit** the port to be logged
- Specify the **Logging Level** of for each port as:

Level 0 Turns off logging for the selected port
Level 1 Logs all User connection events to the port
Level 2 Logs all data transferred to and from the port and all changes in hardware flow control status and all User connection events
Level 3 Logs all data transferred from the port and all changes in hardware flow control status and all User connection events
Level 4 Logs all data tran*sferred to the port and all changes in hardware flow control status and all User connection events*

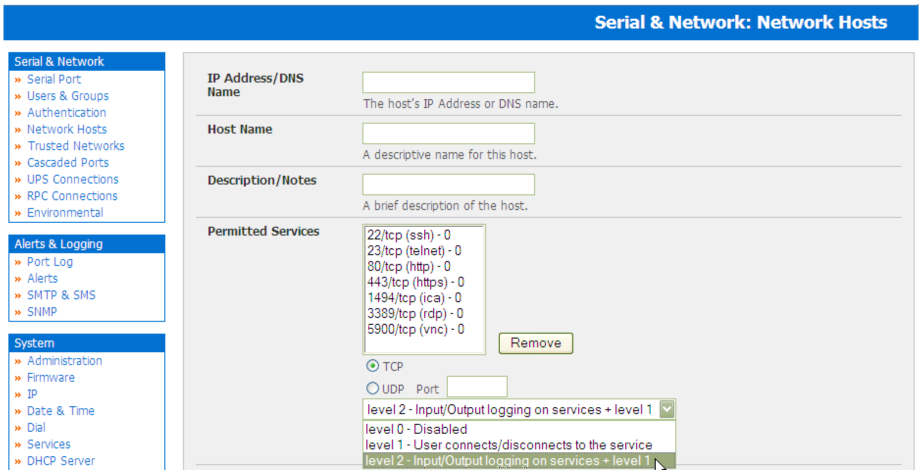
- Click **Apply**

Note**: A cache of the most recent 8K of logged data per serial port is maintained locally (in addition to the Logs which are transmitted for remote/USB flash storage). To view the local cache of logged serial port data select **Manage: Port Logs

7.5 Network TCP or UDP Port Logging

The Console Servers can also log any access to and communications with network attached Hosts.

- For each Host, when you set up the Permitted Services which are authorized to be used, you also must set up the level of logging that is to be maintained for each service



IP Address/DNS Name
The host's IP Address or DNS name.

Host Name
A descriptive name for this host.

Description/Notes
A brief description of the host.

Permitted Services
22/tcp (ssh) - 0
23/tcp (telnet) - 0
80/tcp (http) - 0
443/tcp (https) - 0
1494/tcp (ica) - 0
3389/tcp (rdp) - 0
5900/tcp (vnc) - 0

● TCP
○ UDP Port

level 2 - Input/Output logging on services + level 1
level 0 - Disabled
level 1 - User connects/disconnects to the service
level 2 - Input/Output logging on services + level 1

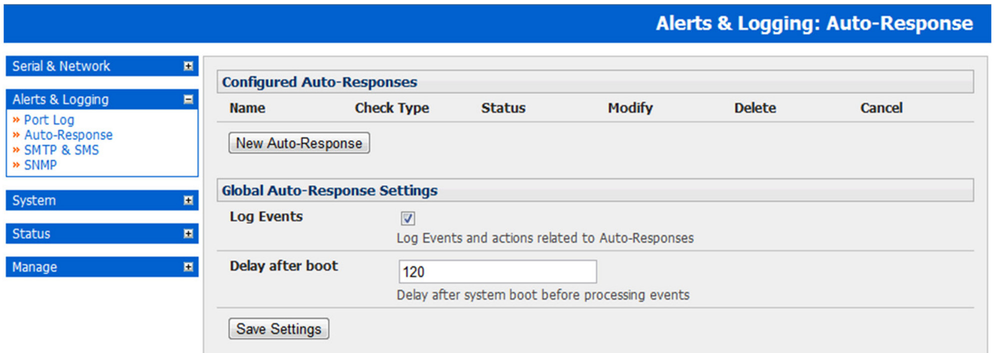
- Specify the logging level that is to be maintained for that particular TDC/UDP port/service on that particular Host:

Level 0 Turns off logging for the selected TDC/UDP port to the selected Host
Level 1 Logs all connection events to the port
Level 2 Logs all data transferred to and from the port

- Click **Add** then click **Apply**

7.6 Auto-Response Event Logging

- Check Log Events on Alerts & Logging: Auto-Response to enable logging all Auto-Response activities



Configured Auto-Responses

Name	Check Type	Status	Modify	Delete	Cancel

Global Auto-Response Settings

Log Events ☑
Log Events and actions related to Auto-Responses

Delay after boot 120
Delay after system boot before processing events

7.7 Power Device Logging

The Console Server also logs access and communications with network attached hosts and maintains a history of the UPS and PDU power status.

To activate and set the desired levels of logging for each serial (Section 7.4) and/or network port (Section 7.5) and/or power and environment UPS (refer Chapter 8).

Chapter 8: Power and Environment

The B094-008-2E-M-F, B094-008-2E-V, B095-004/003 and B092-016 Console Servers and B096-048/032/016 Console Server Management Switch products embed software that can be used to manage connected Power Distribution Systems (PDU's), IPMI devices and Uninterruptible Power Supplies (UPS's) supplied by a number of vendors, and some the environmental monitoring devices. B092-016 Console Server with PowerAlert also embeds Tripp Lite's PowerAlert software.

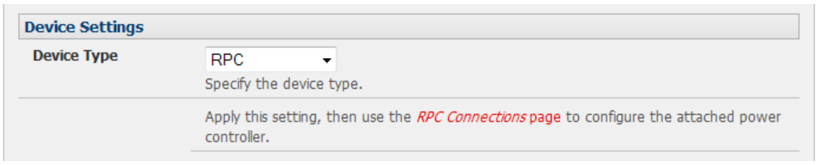
8.1 Remote Power Control (RPC)

The Console Server Management Console monitors and controls Remote Power Control (RPC) devices using the embedded PowerMan and NUT open source management tool. RPC's include power distribution units (PDU's) and IPMI power devices.

8.1.1 RPC connection

Serial and network connected RPC's must first be connected to, and configured to communicate with, the Console Server:

- For serial RPC's, connect the PDU to the selected serial port on the Console Server. From the **Serial and Network: Serial Port** menu, configure the **Common Settings** of that port with the RS232 properties required by the PDU (refer to *Chapter 4.1.1 Common Settings*). Then select **RPC** as the **Device Type**
- Similarly for each network connected RPC, go to **Serial & Network: Network Hosts** menu and configure the RPC as a connected Host



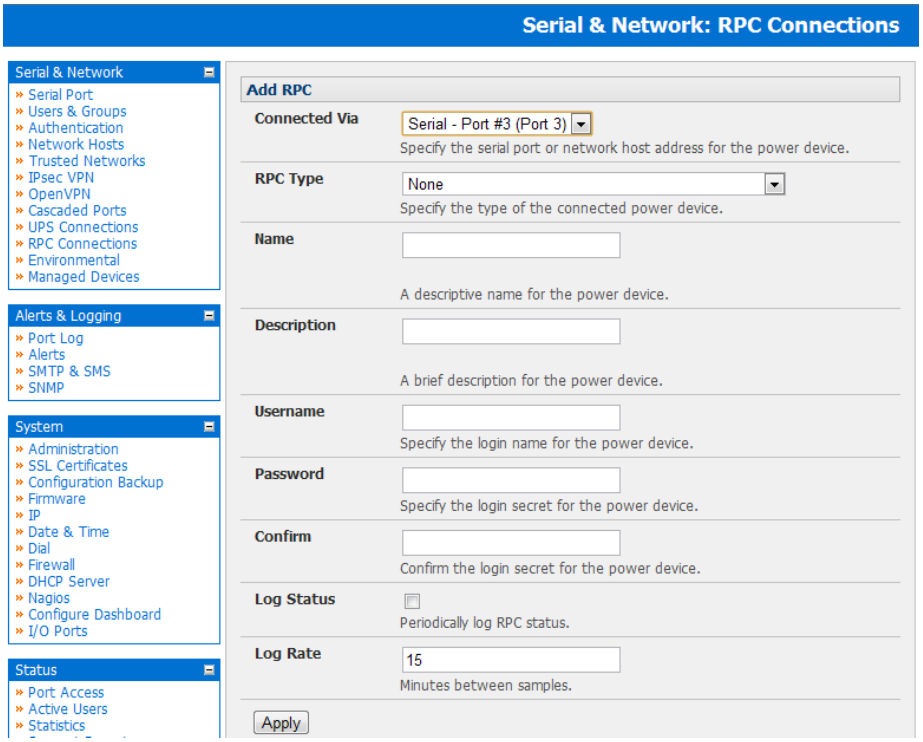
Device Settings

Device Type: RPC

Specify the device type.

Apply this setting, then use the *RPC Connections* page to configure the attached power controller.

- Select the **Serial & Network: RPC Connections** menu. This will display all the RPC connections that have already been configured
- Click **Add RPC** In **Connected Via**, select the pre-configured serial port or the network host address that connects to the RPC
- Enter a **RPC Name** and **Description** for the RPC
- Enter the **Username** and **Password** used to login into the RPC. Note that these login credentials are not related the Users and access privileges you will have configured in *Serial & Networks: Users & Groups*
- Check **Log Status** and specify the **Log Rate** (minutes between samples) if you wish the status from this RPC to be logged. These logs can be views from the **Status: RPC Status** screen
- Click **Apply**



Serial & Network: RPC Connections

Add RPC

Connected Via Serial - Port #3 (Port 3)
Specify the serial port or network host address for the power device.

RPC Type None
Specify the type of the connected power device.

Name
A descriptive name for the power device.

Description
A brief description for the power device.

Username
Specify the login name for the power device.

Password
Specify the login secret for the power device.

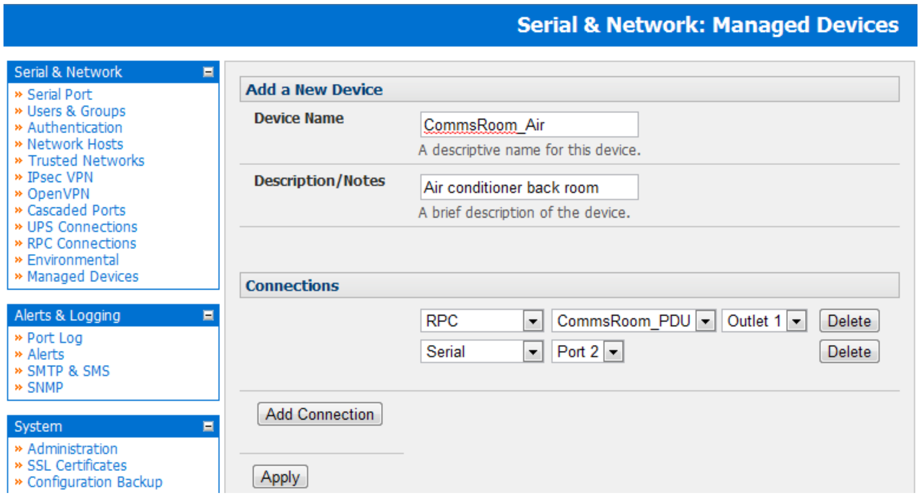
Confirm
Confirm the login secret for the power device.

Log Status □
Periodically log RPC status.

Log Rate 15
Minutes between samples.

Now you have set up a new serially or network connected RPC device, this will automatically create a corresponding new Managed Device with the same Name /Description as the RPC The outlet names on the RPC/PDU Managed Device will by default be “Outlet 1” “Outlet 2”.

You can now establish a “connection” between particular Managed Device that draws power from the particular RPC/PDU outlet (using **Serial & Network: Managed Devices** - refer *Chapter 4*). The outlet will then take up the name of the powered Managed Device.



Serial & Network: Managed Devices

Add a New Device

Device Name CommsRoom_Air
A descriptive name for this device.

Description/Notes Air conditioner back room
A brief description of the device.

Connections

RPC CommsRoom_PDU Outlet 1

Serial Port 2

Note*: The Management Console has support for a number of network and serial PDU’s. If your PDU is not on the default list, it is simple to add support for more devices. This is covered in Chapter 14: Advanced Configurations*

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, the Administrator first enters the IP address/domain name of the BMC or service processor (e.g. a Dell DRAC) in ***Serial & Network: Network Hosts****. Then in* ***Serial & Network: RPC Connections****, the Administrator specifies the* ***RPC Type*** *to be IPMI1.5 or 2.0*

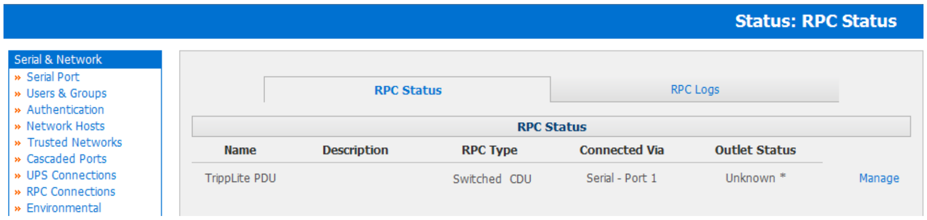
8.1.2 RPC alerts

You can now set PDU and IPMI alerts using **Alerts & Logging: Alerts** (refer to *Chapter 7*)

8.1.3 RPC status

You can monitor the current status of your network and serially connected PDU's and IPMI RPC's

- Select the **Status: RPC Status** menu. A table with the summary status of all connected RPC hardware will be displayed



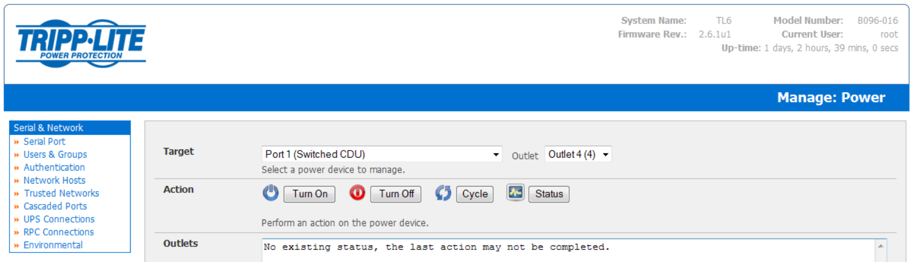
RPC Status

Name	Description	RPC Type	Connected Via	Outlet Status	
TrippLite PDU		Switched CDU	Serial - Port 1	Unknown *	Manage

- Click on **View Log** or select the **RPC Logs** menu. You will be presented with a table of the history and detailed graphical information on the select RPC
- Click **Manage** to query or control the individual power outlet. This will take you to the **Manage: Power** screen

8.1.4 User power management

The Power Manager enables both Users and Administrators to access and control the configured serial and network attached PDU power strips, and servers with embedded IPMI service processors or BMC's:



System Name: TL6 Model Number: B096-016
Firmware Rev.: 2.6.1u1 Current User: root
Up-time: 1 days, 2 hours, 39 mins, 0 secs

Target: Port 1 (Switched CDU) Outlet: Outlet 4 (4)
Select a power device to manage.

Action
Perform an action on the power device.

Outlets

```
No existing status, the last action may not be completed.
```

- Select the **Manage: Power** and the particular **Target** power device to be controlled (or click **Manage** on the **Status: RPC Status** menu)
- The outlet status is displayed. You can initiate the desired **Action** to be taken by selecting the appropriate icon:

Power ON

Power OFF

Power Cycle

Power Status

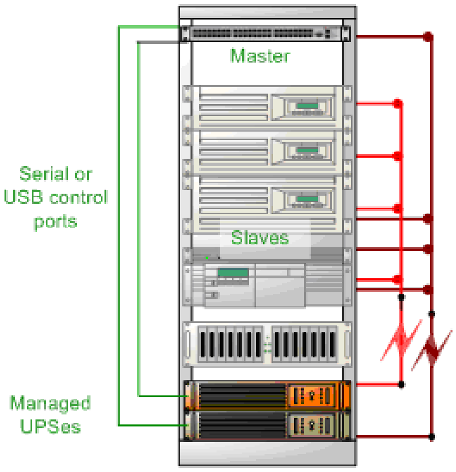
You will only be presented with icons for those operations that are supported by the **Target** you have selected

8.2 Uninterruptible Power Supply Control (UPS)

The Console Servers manage UPS hardware using Network UPS Tools (refer Section 8.2.6 for an overview of embedded open source Network UPS Tools - NUT software)

8.2.1 Managed UPS connections

A **Managed UPS** is a UPS that is connected by serial or USB cable or by the network to the Console Server. 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 its status and take appropriate action (such as shutdown in event of low battery).

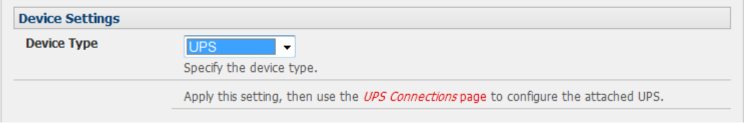


The Console Server may or may not be drawing power through the Managed UPS (see the *Configure UPS powering the Console Server* section below).

When the UPS's battery power reaches critical, the Console Server signals and waits for Slaves to shutdown, then powers off the UPS.

Serial and network connected UPS's must first be configured on the Console Server with the relevant serial control ports reserved for UPS usage, or with the UPS allocated as a connected Host:

- Select **UPS** as the Device Type in the **Serial & Network: Serial Port** menu for each port which has Master control over a UPS and in the **Serial & Network: Network Hosts** menu for each network connected UPS (refer to *Chapter 4*)



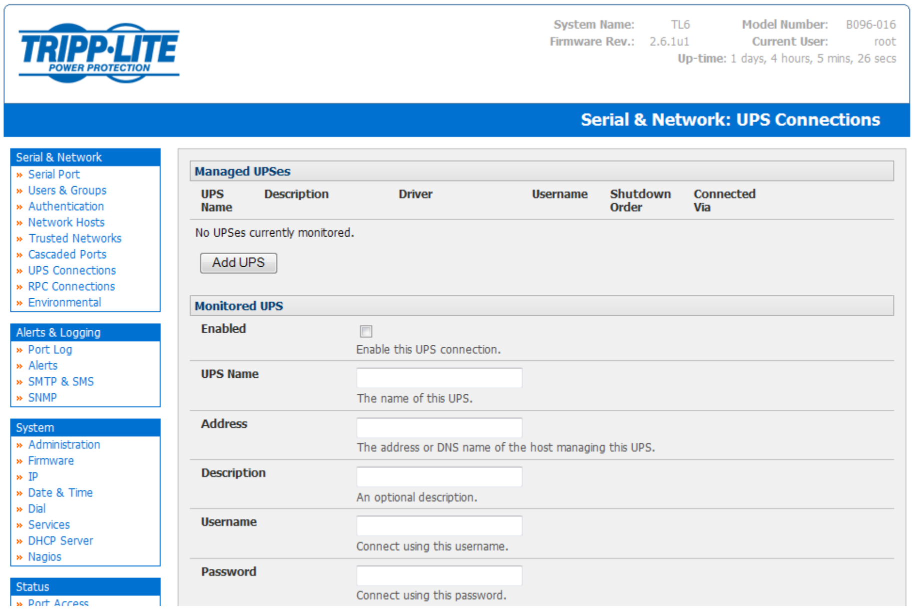
Device Settings

Device Type UPS

Specify the device type.

Apply this setting, then use the *UPS Connections* page to configure the attached UPS.

No such configuration is required for USB-connected UPS hardware.



System Name: TL6 Model Number: B096-016
Firmware Rev.: 2.6.1u1 Current User: root
Up-time: 1 days, 4 hours, 5 mins, 26 secs

Serial & Network: UPS Connections

Managed UPSes

UPS Name	Description	Driver	Username	Shutdown Order	Connected Via

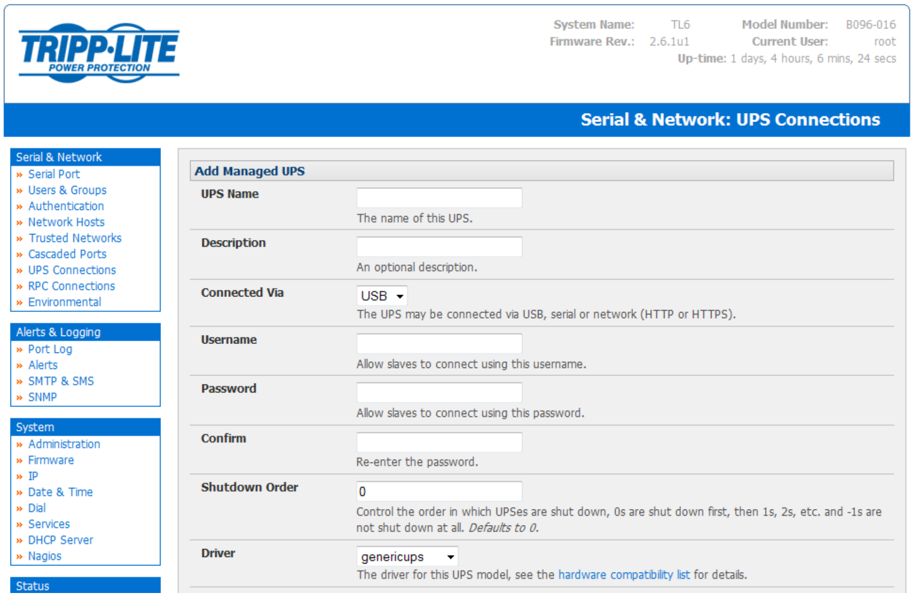
No UPSes currently monitored.

Monitored UPS

Enabled	□ Enable this UPS connection.
UPS Name	The name of this UPS.
Address	The address or DNS name of the host managing this UPS.
Description	An optional description.
Username	Connect using this username.
Password	Connect using this password.

- Select the **Serial & Network: UPS Connections** menu. The **Managed UPSes** section will display all the UPS connections that have already been configured.

- Click **Add UPS**



System Name: TL6 Model Number: B096-016
Firmware Rev.: 2.6.1u1 Current User: root
Up-time: 1 days, 4 hours, 6 mins, 24 secs

Serial & Network: UPS Connections

Add Managed UPS

UPS Name	The name of this UPS.
Description	An optional description.
Connected Via	USB The UPS may be connected via USB, serial or network (HTTP or HTTPS).
Username	Allow slaves to connect using this username.
Password	Allow slaves to connect using this password.
Confirm	Re-enter the password.
Shutdown Order	0 Control the order in which UPSes are shut down, 0s are shut down first, then 1s, 2s, etc. and -1s are not shut down at all. *Defaults to 0.*
Driver	genericups The driver for this UPS model, see the hardware compatibility list for details.

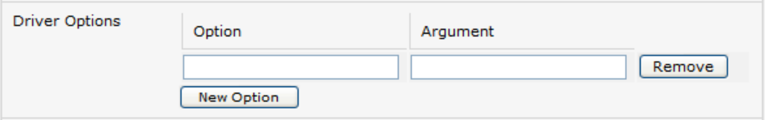
- Enter a **UPS Name** and **Description** (optional) and identify if the UPS will be **Connected Via** USB or over pre-configured serial port or via HTTP/HTTPS over the preconfigured network Host connection

- Enter the UPS login details. This **Username** and **Password** is used by Slaves of this UPS (i.e. other computers that are drawing power through this UPS) to connect to the Console Server for monitoring of the UPS status and shutdown when battery power is low. Monitoring will typically be performed using the *upsmon* client running on the Slave server. See *section 8.5.4* for details on setting up *upsmon* on Slave servers powered by the UPS

Note**: These login credentials are not related to the Users and access privileges you will have configured in **Serial & Networks: Users & Groups

- If you have multiple UPS's and require them to be shut down in a specific order, specify the **Shutdown Order** for this UPS. This is a positive whole number, or *-1. 0s* are shut down first, then *1s, 2s*, etc. *-1s* are not shut down at all. Defaults to *0*

- Select the **Driver** that will be used to communicate with the UPS. The drop-down menu presents a full selection of drivers from the latest Network UPS Tools (NUT version 2.2.0) and additional information on compatible UPS hardware can be found at http://www.networkupstools.org/compat/stable.html
- Click **New Options** in **Driver Options** if you need to set driver-specific options for your selected NUT driver and hardware combination (more details at http://www.networkupstools.org/doc



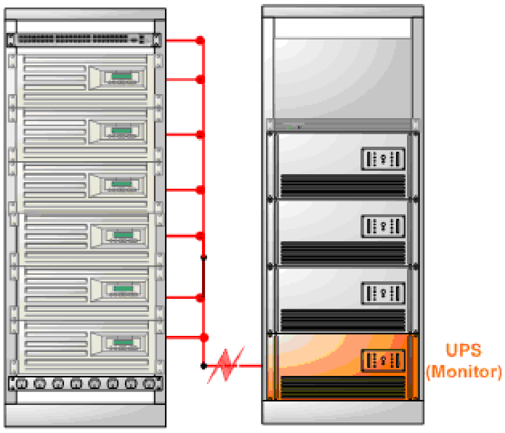
- Check **Log Status** and specify the **Log Rate** (i.e. minutes between samples) if you wish the status from this UPS to be logged. These logs can be views from the **Status: UPS Status screen**
- Check **Enable Nagios** to enable this UPS to be monitored using Nagios central management
- Click **Apply**

You can also customize the *upsmon*, *upsd* and *upsc* settings for this UPS hardware directly from the command line

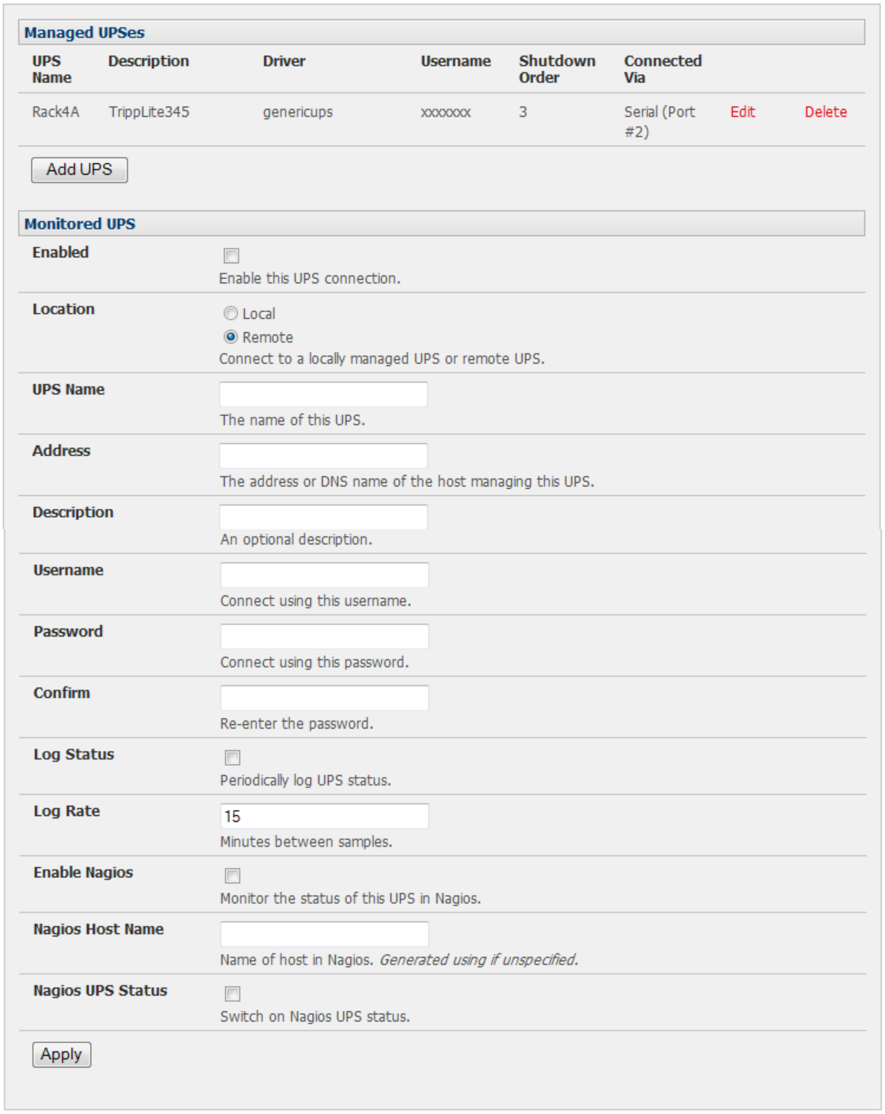
8.2.2 Configure UPS powering the Console Server

A **Monitored UPS** is a UPS that is providing the power to the Console Server. The purpose of configuring a Monitored UPS is to provide an opportunity to perform any "last gasp" actions before power is lost during a power failure. This is achieved by placing a script in */etc/config/scripts/ups-shutdown*. You may use the */etc/scripts/ups-shutdown* as a template. This script is run when then UPS reaches critical battery status.

- If the Console Server is drawing power through a Managed UPS that has already been configured, select **Local**, enter the Managed **UPS Name** and check **Enabled**. The Console Server continues to be the master of this UPS



- If the UPS that powers the Console Server is not a Managed UPS for that Console Server, then the Console Server can still connect to a remote NUT server (*upsd*) to monitor its status as a Slave. In this case, select **Remote**, and enter the address, username and password to connect.



Managed UPSes

UPS Name	Description	Driver	Username	Shutdown Order	Connected Via		
Rack4A	TrippLite345	genericups	xxxxxxx	3	Serial (Port #2)	Edit	Delete

Monitored UPS

Enabled	□ Enable this UPS connection.
Location	○ Local ● Remote Connect to a locally managed UPS or remote UPS.
UPS Name	The name of this UPS.
Address	The address or DNS name of the host managing this UPS.
Description	An optional description.
Username	Connect using this username.
Password	Connect using this password.
Confirm	Re-enter the password.
Log Status	□ Periodically log UPS status.
Log Rate	15 Minutes between samples.
Enable Nagios	□ Monitor the status of this UPS in Nagios.
Nagios Host Name	Name of host in Nagios. *Generated using if unspecified.*
Nagios UPS Status	□ Switch on Nagios UPS status.

8.2.3 Configuring powered computers to monitor a Managed UPS

Once you have added a Managed UPS, each server that is drawing power through the UPS should be setup to monitor the UPS status as a Slave. This is done by installing the NUT package on each server, and setting up *upsmon* to connect to the Console Server.

Refer to the NUT documentation for details on how this is done, specifically sections 13.5 to 13.10. http://eu1.networkupstools.org/doc/2.2.0/INSTALL.html

An example upsmon.conf entry might look like:

- *MONITOR managedups@192.168.0.1 1 username password Slave*
- *managedups* is the UPS Name of the Managed UPS
- *192.168.0.1* is the IP address of the Console Server
- *1* indicates the server has a single power supply attached to this UPS
- *username* is the Username of the Managed UPS
- *password* is the Password of the Manager UPS

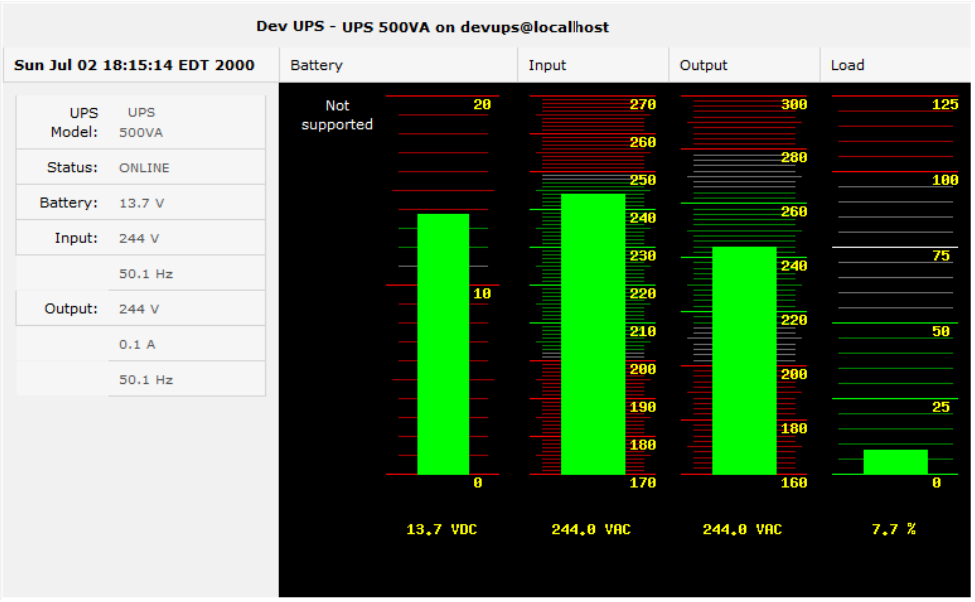
8.2.4 UPS alerts

You can now set UPS alerts using **Alerts & Logging: Alerts** (refer to *Chapter 7*)

8.2.5 UPS status

You can monitor the current status of all your Managed or Monitored UPS's, whether they are on the network or connected serially or via USB:

- Select the **Status: UPS Status** menu and a table with the summary status of all connected UPS hardware will be displayed
- Click on any particular UPS **System** name in the table and you will be presented with a more detailed graphical information on the select UPS System



Dev UPS - UPS 500VA on devups@localhost

Sun Jul 02 18:15:14 EDT 2000

UPS Model:	UPS 500VA
Status:	ONLINE
Battery:	13.7 V
Input:	244 V
	50.1 Hz
Output:	244 V
	0.1 A
	50.1 Hz

- Click on any particular **All Data** for any UPS System in the table for more status and configuration information on the select UPS System
- Select **UPS Logs** and you will be presented with the log table of the load, battery charge level, temperature and other status information from all the Managed and Monitored UPS systems. This information will be logged for all UPS's which were configured with Log Status checked. The information is also presented graphically

8.2.6 Overview of Network UPS Tools (NUT)

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 which are connected.

NUT can be configured using the Management Console as described above, or you can configure the tools and manage the UPS's directly from the command line. This section provides an overview of NUT. You can find full documentation at http://www.networkupstools.org/doc.

NUT is built on a networked model with a layered scheme of drivers, server and clients.

1. 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 they understand the specific language of each UPS and map it back to a compatibility layer. This means both an expensive "smart" protocol UPS and a simple "power strip" model can be handled transparently.
2. 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 UPS's and can then serve this status data to many clients. *upsd* also contains access control features to limit the abilities of the clients (so only authorized hosts may monitor or control the UPS hardware).
3. 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 ***upsmon*** client enables servers that draw power through the UPS (i.e. Slaves of the UPS) to shutdown gracefully when the battery power reaches critical. Additionally, one server is designated the Master of the UPS, and is responsible for shutting down the UPS itself when all Slaves have shut down. Typically, the Master of the UPS is the one connected to the UPS via serial or USB cable.

   upsmon can monitor multiple UPS's, so high-end servers which receive power from multiple UPS's simultaneously won't initiate a shutdown until the total power situation across all source UPS's becomes critical.

   There also the two status/logging clients, ***upsc*** and ***upslog***. The *upsc* client provides a quick way to poll the status of a UPS. It can be used inside shell scripts and other programs that need UPS status information. *upslog* is a background service that periodically polls the status of a UPS, writing it to a file.

   All these clients run on the Console Server (for Management Console presentations) but they also are run remotely (on locally powered servers and remote monitoring systems).

This layered NUT architecture enables:

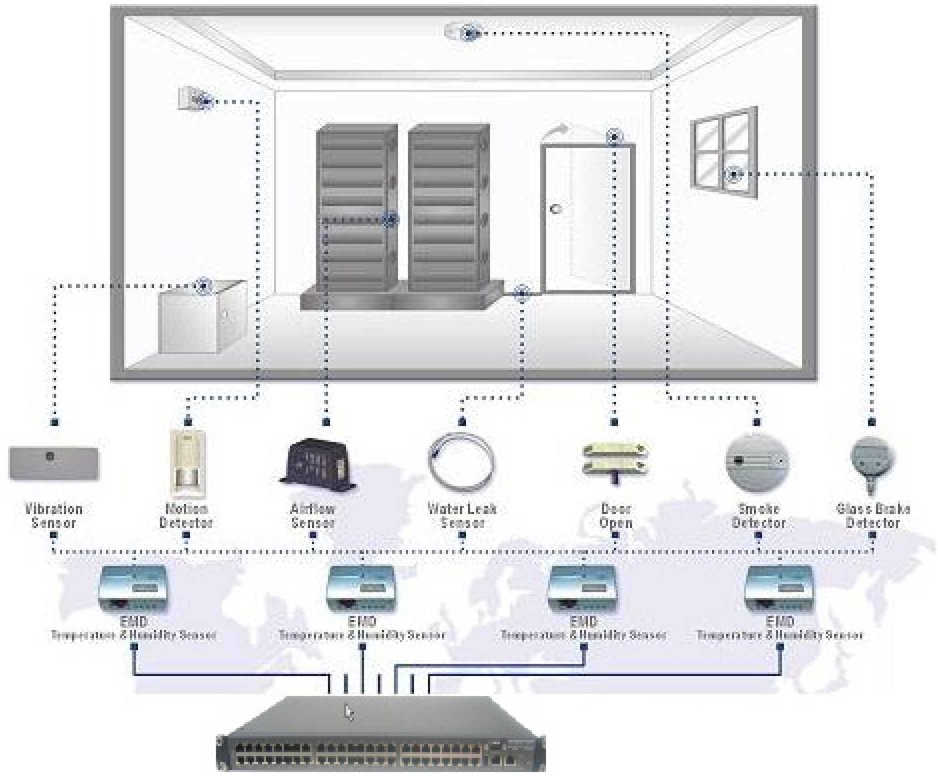
- Multiple architecture support: NUT can manage serial and USB-connected models with the same common interface. SNMP equipment can also be monitored (although at this stage this is still pre-release with experimental drivers and this feature will be added to the embedded UPS tools in future release).
- Multiple clients monitoring one UPS: Multiple systems may monitor a single UPS using only their network connections. There's a wide selection of client programs which support monitoring UPS hardware via NUT (Big Sister, Cacti, Nagios, Windows and more). Refer to *www.networkupstools.org/client-projects*.)

So NUT supports the more complex power architectures found in data centers, computer rooms and NOCs where many UPS's from many vendors power many systems with many clients and each of the larger UPS's power multiple devices and many of these devices are themselves dual powered.

8.3 Environmental Monitoring

The Environmental Monitoring Device (EMD), model B090-EMD, can be connected to any Console Server serial port and each Console Server can support multiple EMD's. Each EMD has one temperature and one humidity sensor and one general purpose status sensor which can be connected to a smoke detector, water detector, vibration or open-door sensor.
The B095-004/003 Console Server models also each has an internal temperature sensor.

Using the Management Console, Administrators can view the ambient temperature and humidity and set the EMD to automatically send alarms progressively from warning levels to critical alerts.


Vibration Sensor
Motion Detector
Airflow Sensor
Water Leak Sensor
Door Open
Smoke Detector
Glass Brake Detector
EMD Temperature & Humidity Sensor
EMD Temperature & Humidity Sensor
EMD Temperature & Humidity Sensor
EMD Temperature & Humidity Sensor

8.3.1 Connecting the EMD

The Environmental Monitoring Sensor (EMD) connects to any serial port on the Console Server via a special EMD Adapter and standard CAT5 cable. The EMD is powered over this serial connection and communicates using a custom handshake protocol. It is not an RS232 device and should not be connected without the adapter:

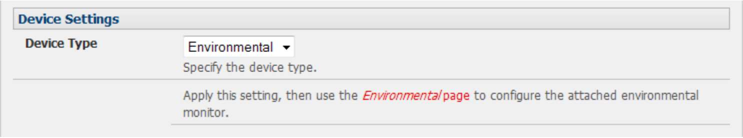
- Plug the RJ plug on the EMD Adapter (model B090-EMD-ADP) into RJ45 Port on the EMD (model B090-EMD). Then connect the Console Server serial port to the RJ45 port of the EMD Adapter using the provided UTP cable. If the 6 foot (2 meter) UTP cable provided with the EMD is not long enough it can be replaced with a standard Cat5 UTP cable up to 33 feet (10meters) in length (Tripp Lite N002 series cables)

- Screw the bare wires on any smoke detector, water detector, vibration sensor, open-door sensor or general purpose open/close status sensors into the terminals on the EMD:

o	B090-WLS	Console Server Water Leak Sensor
o	B090-DCS	Console Server Door Contact Sensor
o	B090-VS	Console Server Vibration Sensor
o	B090-SD-110	Console Server Smoke Detector - 110V
o	B090-SD-220	Console Server Smoke Detector - 220V

The EMD can be used only with a Console Server and cannot be connected to standard RS232 serial ports on other appliances.

- Select **Environmental** as the **Device Type** in the **Serial & Network: Serial Port** menu for the port to which the EMD is to be attached. No particular Common Settings are required.
- Click **Apply**



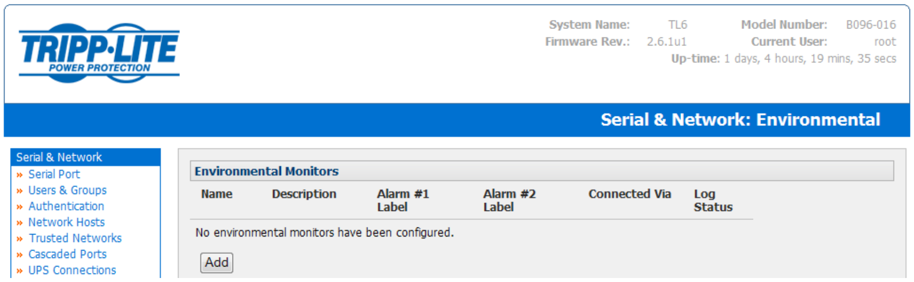
Device Settings

Device Type: Environmental

Specify the device type.

Apply this setting, then use the *Environmental* page to configure the attached environmental monitor.

- Select the **Serial & Network: Environmental** menu. This will display all the EMD connections that have already been configured



System Name: TL6 Model Number: B096-016
Firmware Rev.: 2.6.1u1 Current User: root
Up-time: 1 days, 4 hours, 19 mins, 35 secs

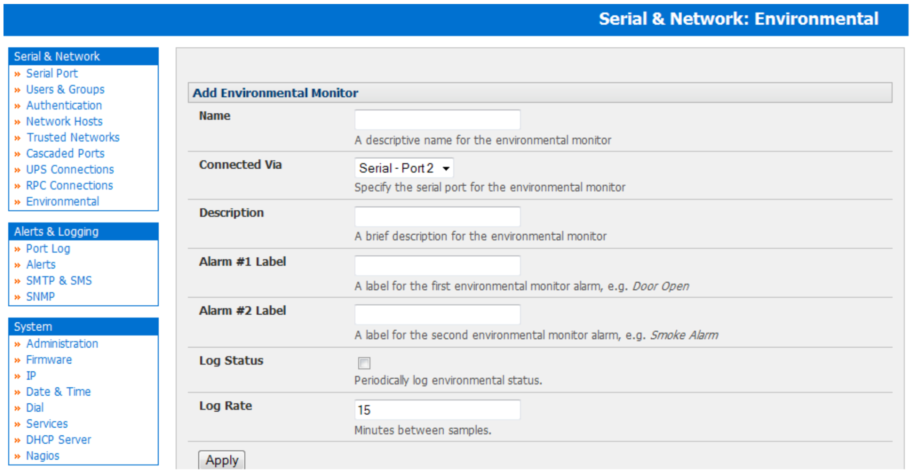
Serial & Network: Environmental

Environmental Monitors

Name	Description	Alarm #1 Label	Alarm #2 Label	Connected Via	Log Status

No environmental monitors have been configured.

- Click **Add**



Add Environmental Monitor

Name	A descriptive name for the environmental monitor
Connected Via	Serial - Port 2 Specify the serial port for the environmental monitor
Description	A brief description for the environmental monitor
Alarm #1 Label	A label for the first environmental monitor alarm, e.g. *Door Open*
Alarm #2 Label	A label for the second environmental monitor alarm, e.g. *Smoke Alarm*
Log Status	☐ Periodically log environmental status.
Log Rate	15 Minutes between samples.

- Enter a **Name** and **Description** for the EMD and select pre-configured serial port that the EMD will be **Connected Via**
- Provide **Labels** for each of the two alarms
- Check **Log Status** and specify the **Log Rate** (minutes between samples) if you wish the status from this EMD to be logged. These logs can be views from the **Status: Environmental Status** screen
- Click **Apply**

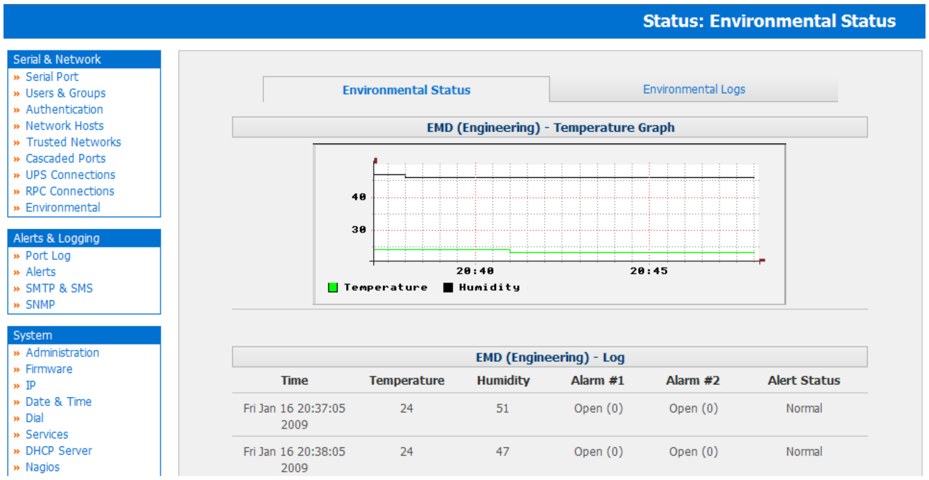
8.3.2 Environmental alerts

You can now set temperature, humidity and probe status alerts using **Alerts & Logging: Alerts** (refer to *Chapter 7*)

8.3.3 Environmental status

You can monitor the current status of all of EMDs and their probes

- Select the **Status: Environmental Status** menu and a table with the summary status of all connected EMD hardware will be displayed
- Click on **View Log** or select the **Environmental Logs** menu and you will be presented with a table and graphical plot of the log history of the select EMD



EMD (Engineering) - Log

Time	Temperature	Humidity	Alarm #1	Alarm #2	Alert Status
Fri Jan 16 20:37:05 2009	24	51	Open (0)	Open (0)	Normal
Fri Jan 16 20:38:05 2009	24	47	Open (0)	Open (0)	Normal

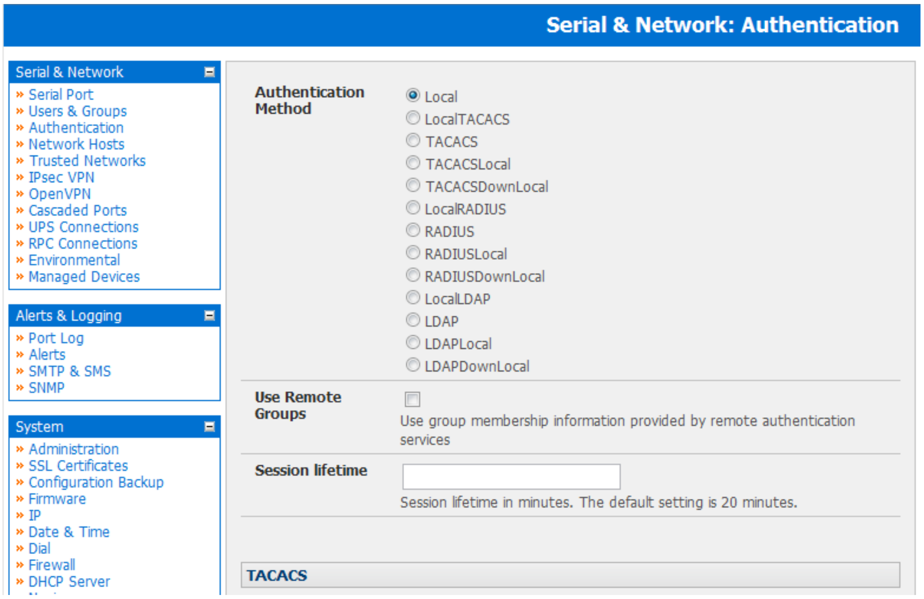
Chapter 9: Authentication

The Tripp Lite Console Server is a dedicated Linux computer, and it embodies popular and proven Linux software modules for secure network access (OpenSSH) and communications (OpenSSL) and sophisticated user authentication (PAM, RADIUS, TACACS+, Kerberos and LDAP).

- 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 to establish a secure Administration connection to the Console Server

9.1 Authentication Configuration

Authentication can be performed locally, or remotely using an LDAP, Radius or TACACS+ authentication server. The default authentication method for the Console Server is Local.



Authentication Method

● Local
○ LocalTACACS
○ TACACS
○ TACACSLocal
○ TACACSDownLocal
○ LocalRADIUS
○ RADIUS
○ RADIUSLocal
○ RADIUSDownLocal
○ LocalLDAP
○ LDAP
○ LDAPLocal
○ LDAPDownLocal

Use Remote Groups □
Use group membership information provided by remote authentication services

Session lifetime
Session lifetime in minutes. The default setting is 20 minutes.

Any authentication method that is configured will be used for authentication of any user attempting 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** Kerberos or **LDAP**) 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)

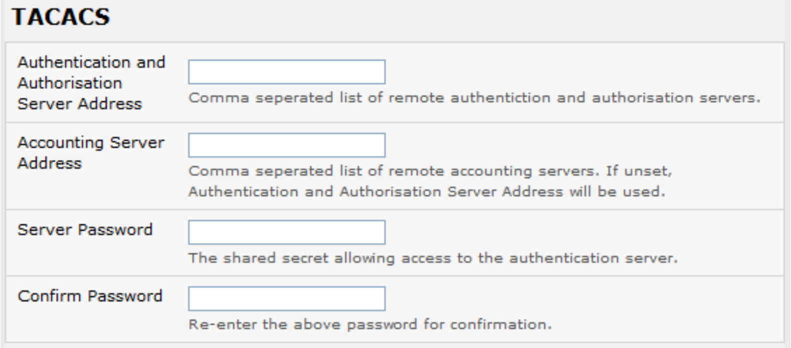
9.1.1 Local authentication

- Select **Serial and Network: Authentication** and check **Local**
- Click **Apply**

9.1.2 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** and check **TACAS** or **LocalTACACS** or **TACACSLocal** or **TACACSDownLocal**



TACACS

Authentication and Authorisation Server Address	Comma seperated list of remote authentiction and authorisation servers.
Accounting Server Address	Comma seperated list of remote accounting servers. If unset, Authentication and Authorisation Server Address will be used.
Server Password	The shared secret allowing access to the authentication server.
Confirm Password	Re-enter the above password for confirmation.

- 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.
- In addition to multiple remote servers, you can also enter for separate lists of Authentication/Authorization servers and Accounting servers. If no Accounting servers are specified, the Authentication/Authorization servers are used instead.
- Enter the **Server Password**

  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: *The console server normally interprets a user with a TACACS* priv-lvl *of 12 or above as an* admin *user. There is also a special privilege level 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 (i.e., checked in the user interface), there are no escalations of privileges based on the* priv-lvl *value from the TACACS server.*

Also note that if the only privilege level configured for one or more TACACS users is the priv-lvl *(e.g., no specific port access or group memberships set) level, you will revoke access to the console server for those users with whom this level is enabled. Users will not be a member of any group, even if the Retrieve Remote groups option in the Authentication menu is enabled.*

- Click **Apply**. TACAS+ remote authentication will now be used for all user access to Console Server and serially or network attached devices

TACACS+ The Terminal Access Controller Access Control System (TACACS+) security protocol is a 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. Further information on configuring remote TACACS+ servers can be found at the following sites:

http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a0080094e99.shtml

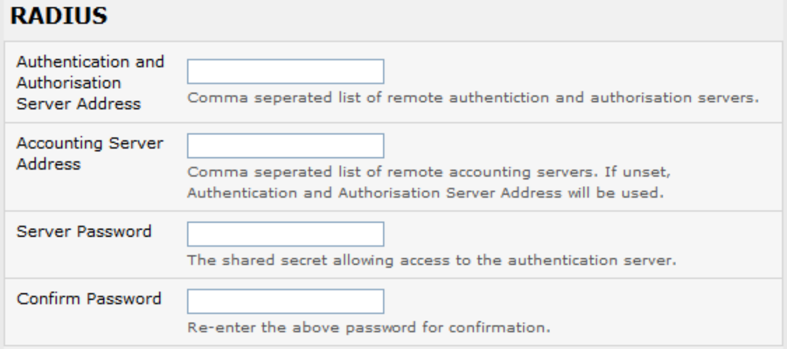
http://www.cisco.com/en/US/products/sw/secursw/ps4911/products_user_guide_chapter09186a00800eb6d6.html

http://cio.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/scprt2/sctplus.htm

9.1.3 RADIUS authentication

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 and Network: Authentication** and check **RADIUS** or **LocalRADIUS** or **RADIUSLocal** or **RADIUSDownLocal**



RADIUS

Authentication and Authorisation Server Address	Comma seperated list of remote authentiction and authorisation servers.
Accounting Server Address	Comma seperated list of remote accounting servers. If unset, Authentication and Authorisation Server Address will be used.
Server Password	The shared secret allowing access to the authentication server.
Confirm Password	Re-enter the above password for confirmation.

- 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
- In addition to multiple remote servers, you can also enter for separate lists of Authentication/Authorization servers and Accounting servers. If no Accounting servers are specified, the Authentication/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

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. Further information on configuring remote RADIUS servers can be found at the following sites:

http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/DepKit/d4fe8248-eecd-49e4-88f6-9e304f97fefc.mspx

http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a00800945cc.shtml

http://www.freeradius.org/

9.1.4 LDAP authentication

With firmware version 3.11 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) follow the common LDAP standards and protocols. Extra steps are required in configuring extra user data (e.g., groups, etc).

The console server may be configured for authentication and authorization of group information from an LDAP server. This group information can be stored in a number of different ways. Active Directory has one method of storage, and OpenLDAP has two methods:

- Active Directory: Each entry for a user has multiple 'memberOf' attributes. Each 'memberOf' value is the full DN of the group they belong to (the user entry will be of objectClass "user").
- OpenLDAP / Posix: Each entry for a user must have a 'gidNumber' attribute. This will be an integer value that functions as the user's primary group (e.g., mapping to the /etc/passwd file within the group ID field). To determine the group, first search for an entry in the directory that contains that group ID. Doing this will also provide the group name (the users are of objectClass "posixAccount" and the groups are of objectClass "posixGroup").
- OpenLDAP / Posix: Each group entry in the group tree (of objectClass 'posixGroup') may have multiple 'memberUid' attributes. These represent secondary groups (e.g., mapping to the /etc/groups file). Each attribute contains a username.

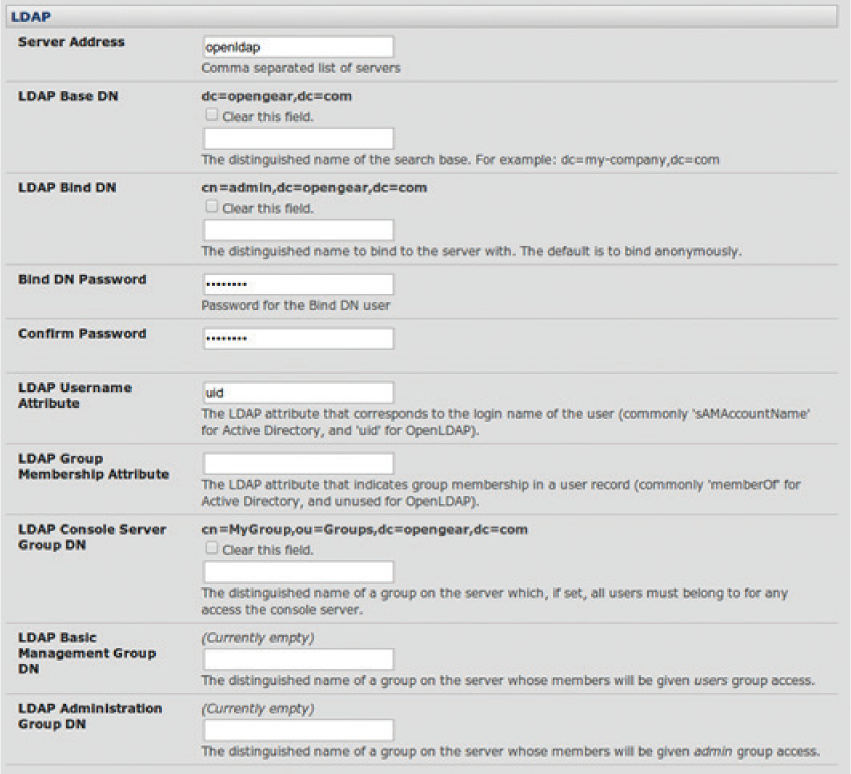
To accommodate these possibilities, the *pam_ldap* module has been modified to perform group queries for each of the three styles. This allows for a 'generic' configuration and does not affect how the LDAP directory is set up.

There are only two parameters that need to be configured based upon a user's search: LDAP username and group membership attributes.

To clarify which 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 below use the following descriptions:

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).



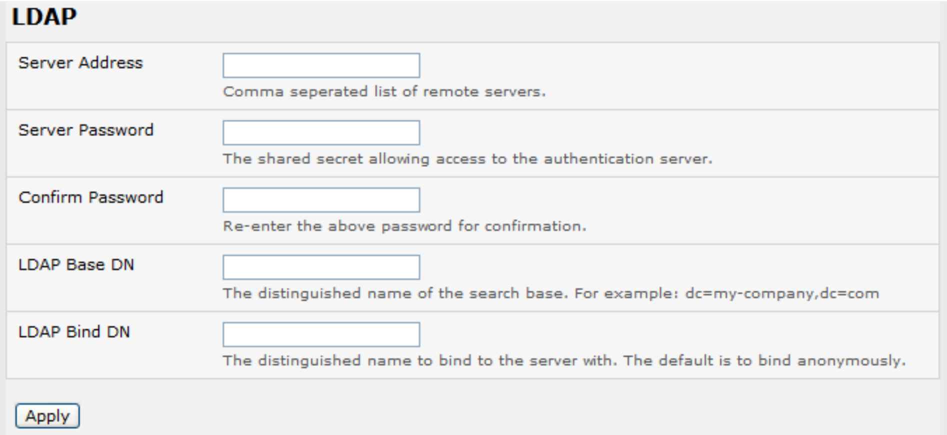
LDAP

Server Address	openldap Comma separated list of servers
LDAP Base DN	dc=opengear,dc=com □ Clear this field. The distinguished name of the search base. For example: dc=my-company,dc=com
LDAP Bind DN	cn=admin,dc=opengear,dc=com □ Clear this field. The distinguished name to bind to the server with. The default is to bind anonymously.
Bind DN Password	•••••••• Password for the Bind DN user
Confirm Password	••••••••
LDAP Username Attribute	uid 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).
LDAP Console Server Group DN	cn=MyGroup,ou=Groups,dc=opengear,dc=com □ Clear this field. The distinguished name of a group on the server which, if set, all users must belong to for any access the console server.
LDAP Basic Management Group DN	*(Currently empty)* The distinguished name of a group on the server whose members will be given *users* group access.
LDAP Administration Group DN	*(Currently empty)* The distinguished name of a group on the server whose members will be given *admin* group access.

Note: *The* libldap *library is particular about ensuring SSL connections using certificates signed by a trusted CA. Setting up a connection to an LDAP server using SSL requires extra attention.*

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:

- Select **Serial and Network: Authentication** and check **LDAP** or **LocalLDAP** or **LDAPLocal** or **LDAPDownLocal**



LDAP

Server Address	Comma seperated list of remote servers.
Server Password	The shared secret allowing access to the authentication server.
Confirm Password	Re-enter the above password for confirmation.
LDAP Base DN	The distinguished name of the search base. For example: dc=my-company,dc=com
LDAP Bind DN	The distinguished name to bind to the server with. The default is to bind anonymously.

- 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.
- Enter the **Server Password**
- Check the **Server Protocol** box if SSL is to be used and/or enforced for communications with the LDAP server. Console servers running firmware v3.11 and above offer three options for LDAPS (LDAP over SSL):
  - o **LDAP over SSL preferred** will attempt to use SSL for authentication, but will fall back to LDAP without SSL if the authentication attempt fails. For example, LDAP over SSL may fail due to certificate errors or the LDAP server may not be contactable on the LDAPS port.
  - o **LDAP over SSL only** will configure the console server to only accept LDAP over SSL. If LDAP over SSL fails, you will only be able to log in to the *console server* as root.
  - o **LDAP (no SSL) only** will configure the console server to only accept LDAP without SSL. If LDAP without SSL fails, you will only be able to log in to the *console server* as root.
- The **Ignore SSL Certificate Error** check box allows you to ignore SSL certificate errors so that LDAP over SSL works regardless of certificate errors. Any certificate can be used—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 that the LDAP server's certificate was signed with onto the *console server*. For example, the LDAP server will contain a certificate singed using the certificate 'myCA.crt'.

Note: *The certificate needs to be in CRT format and* myCA.crt *needs to be installed onto* console server *at* '/etc/config/ldaps_ca.crt'*. The file name must also be* 'ldaps_ca.crt'*. You will need to copy the file and file name manually to this location using 'scp' or:* scp /local/path/to/myCA.crt root@console_server:/etc/config/ldaps_ca.crt

- Click **Apply**. LDAP remote authentication will now be used for all user access to Console Server and serially or network attached devices

LDAP 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. Further information on configuring remote RADIUS servers can be found at the following sites:

http://www.ldapman.org/articles/intro_to_ldap.html

http://www.ldapman.org/servers.html

http://www.linuxplanet.com/linuxplanet/tutorials/5050/1/

http://www.linuxplanet.com/linuxplanet/tutorials/5074/4/

9.1.5 RADIUS/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 configurators unless they are specifically added, at which point they are transformed into a completely local user. The newly added user must authenticate via the remote AAA server, and will not have any access if it is down.

If a local user logs in, they may be authenticated/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 A 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. The user 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 B 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 not have any access.

Example 3:

User C is defined on a RADIUS server only. He has access to all serial ports and network hosts.

Example 4:

User D is locally defined on an appliance using RADIUS for AAA. Even if the user 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.

9.1.6 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, RADIUS and LDAP users can belong to a local group that has been setup 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:

- Select **Serial & Network: Authentication**
- Select the relevant **Authentication Method**
- Check the **Use Remote Groups** button

9.1.7 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:

The above example sets the remote user as a member of testgroup1 and users if groups with those names exist on the Console Server. Any groups which do not exist on the Console Server are ignored.

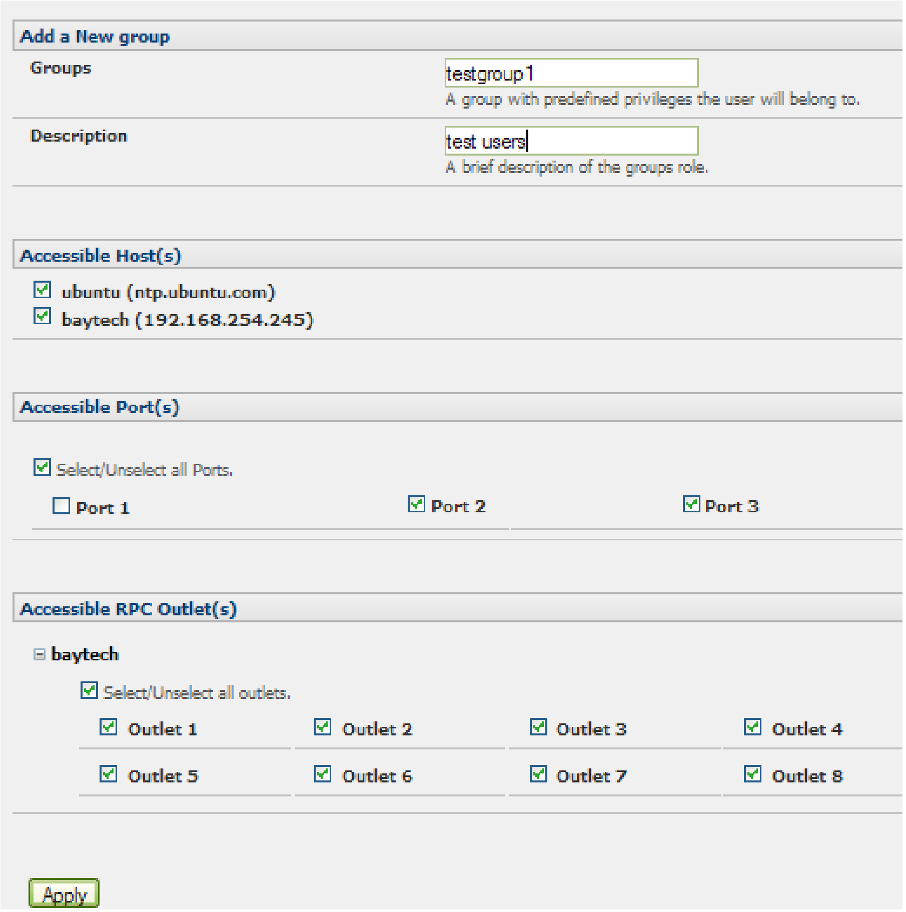
When setting the Framed-Filter-Id, the system may also remove the leading colon for an empty field. To work around this, add some dummy text to the start of the string. For example:

dummy:group_name=testgroup1,users:

- If no group is specified for a user, for example AmandaJones, then the user will have no User Interface and serial port access but limited console 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**



Add a New group

Groups: testgroup1
A group with predefined privileges the user will belong to.

Description: test users
A brief description of the groups role.

Accessible Host(s)

☑ ubuntu (ntp.ubuntu.com)
☑ baytech (192.168.254.245)

Accessible Port(s)

☑ Select/Unselect all Ports.

☐ Port 1 ☑ Port 2 ☑ Port 3

Accessible RPC Outlet(s)

⊟ baytech

☑ Select/Unselect all outlets.

☑ Outlet 1 ☑ Outlet 2 ☑ Outlet 3 ☑ Outlet 4
☑ Outlet 5 ☑ Outlet 6 ☑ Outlet 7 ☑ Outlet 8

9.1.8 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*: Any spaces in the 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:

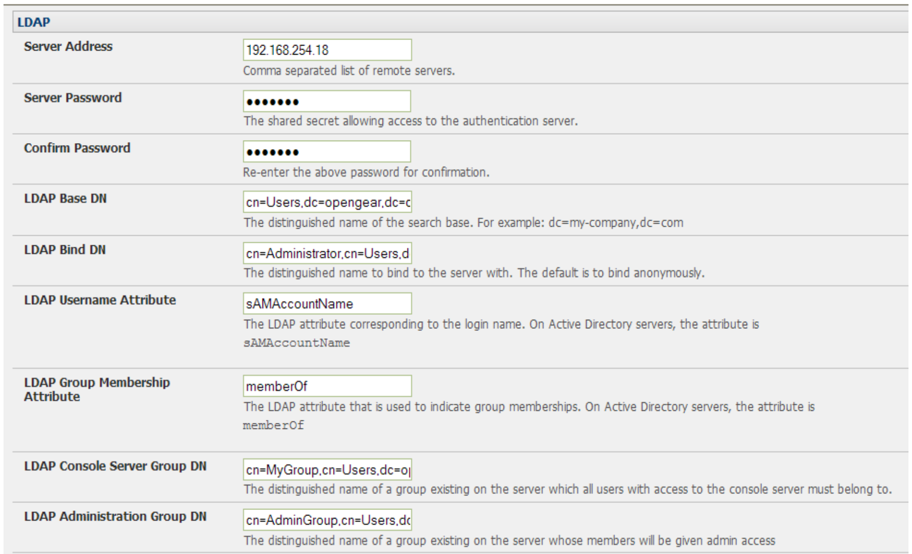
- Complete the fields for standard LDAP authentication including LDAP Server Address, Server Password, LDAP Base DN, LDAP Bind DN and LDAP User Name Attribute
- Enter memberOf for **LDAP Group Membership Attribute** as group membership is currently only supported on Active Directory servers
- If required, enter the group information for **LDAP Console Server Group DN** and/or **LDAP Administration Group DN**

Note: *When using remote groups with LDAP remote auth, you will need to have corresponding local groups on the console server. In situations where LDAP group names can contain upper case and space characters, the local group name on the console server must be all lower case and any spaces must be replaced with underscrores. For example, a remote group on the LDAP server may be* ***My Ldap Access Group****, but needs a corresponding local group on the console server called* ***my_ldap_access_group****. For any group membership to be effective, the local group must specify what the group member is granted access to.*

A user must be a member of the LDAP Console Server Group DN group in order 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 in order 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.

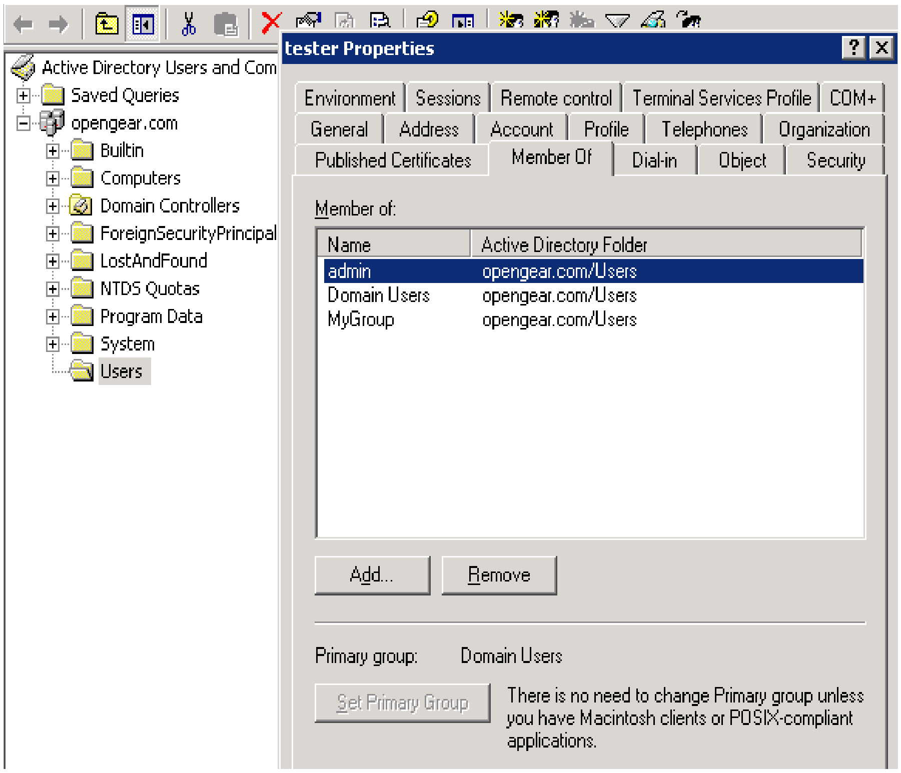
- Click Apply.



LDAP

Server Address	192.168.254.18 Comma separated list of remote servers.
Server Password	••••••• The shared secret allowing access to the authentication server.
Confirm Password	••••••• Re-enter the above password for confirmation.
LDAP Base DN	cn=Users,dc=opengear,dc=c The distinguished name of the search base. For example: dc=my-company,dc=com
LDAP Bind DN	cn=Administrator,cn=Users,d The distinguished name to bind to the server with. The default is to bind anonymously.
LDAP Username Attribute	sAMAccountName The LDAP attribute corresponding to the login name. On Active Directory servers, the attribute is `sAMAccountName`
LDAP Group Membership Attribute	memberOf The LDAP attribute that is used to indicate group memberships. On Active Directory servers, the attribute is `memberOf`
LDAP Console Server Group DN	cn=MyGroup,cn=Users,dc=o The distinguished name of a group existing on the server which all users with access to the console server must belong to.
LDAP Administration Group DN	cn=AdminGroup,cn=Users,dc The distinguished name of a group existing on the server whose members will be given admin access

- Ensure the LDAP service is operational and group names are correct within the Active Directory



Member of:

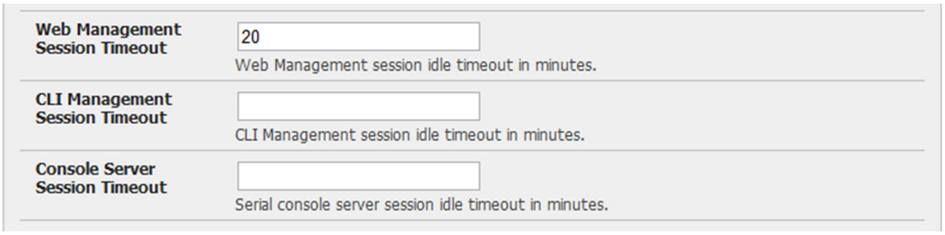
Name	Active Directory Folder
admin	opengear.com/Users
Domain Users	opengear.com/Users
MyGroup	opengear.com/Users

Primary group: Domain Users

There is no need to change Primary group unless you have Macintosh clients or POSIX-compliant applications.

9.1.9 Idle timeout

You can specify amount of time in minutes the console server waits before it terminates an idle ssh, pmshell or web connection.



Web Management Session Timeout	20 Web Management session idle timeout in minutes.
CLI Management Session Timeout	 CLI Management session idle timeout in minutes.
Console Server Session Timeout	 Serial console server session idle timeout in minutes.

- Select Serial and Network: Authentication
- Web Management Session Timeout specifies the browser console session idle timeout in minutes. The default setting is 20 minutes
- CLI Management Session Timeout specifies the ssh console session idle timeout in minutes. The default setting is to never expire
- Console Server Session Timeout specifies the pmshell serial console server session idle timeout in minutes. The default setting is to never expire

9.1.10 Kerberos authentication

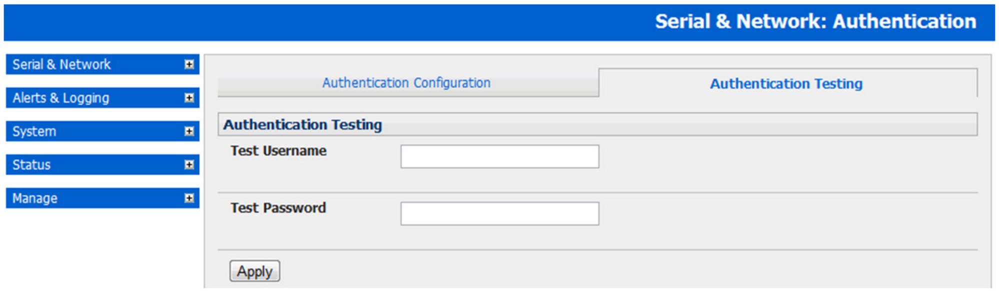
The Kerberos authentication can be used with UNIX and Windows (Active Directory) Kerberos servers. This form of authentication does not provide group information, so a local user with the same username must be created, and permissions set.

***Note**: Kerberos is very sensitive to time differences between the Key Distribution Center (KDC) authentication server and the client device. Please make sure that NTP is enabled, and the time zone is set correctly on the console server.*

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.

9.1.11 Authentication testing

The Authentication Testing tab enables the connection to the remote authentication server to be tested.



9.2 PAM (Pluggable Authentication Modules)

The Console Server supports RADIUS, TACACS+ and LDAP for two-factor authentication via PAM (Pluggable Authentication Modules). PAM is a flexible mechanism for authenticating Users. Nowadays, 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 schemes. These programs need "authentication modules" to be attached to them at run-time in order to work. Which authentication module is to be attached is dependent upon 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:

RADIUS	- pam_radius_auth	(http://www.freeradius.org/pam_radius_auth/)
TACACS+	- pam_tacplus	(http://echelon.pl/pubs/pam_tacplus.html)
LDAP	- pam_ldap	(http://www.padl.com/OSS/pam_ldap.html)

Further modules can be added as required.

Changes may be made to files in /etc/config/pam.d/ which 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 not have any rights, and no password set. They will not appear in the configuration tools.

  Automatically added accounts will not be able to log in if the remote servers are unavailable. RADIUS users are currently assumed to have access to all resources, so will only be authorized to log in to the Console Server. RADIUS users will be authorized each time they access a new resource.

- Admin rights granted over AAA:

  Users may be granted Administrator rights via networked AAA. For TACACS, a priv-lvl of 12 of 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 for both serial ports and host access:

  Permission to access resources may be granted via TACACS by indicating an 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 = xxxxx/port02
      port2 = 192.168.254.145/port05
  }
  global = cleartext mit
}
```

RADIUS Example:

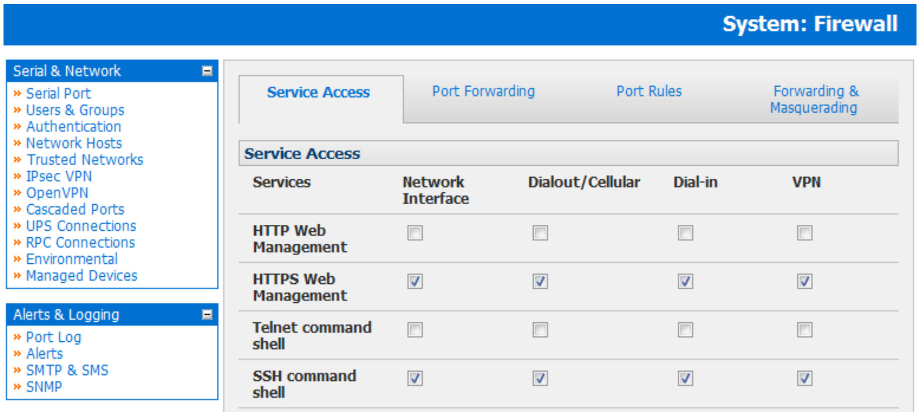
```
paul    Cleartext-Password := "luap"
        Service-Type = Framed-User,
        Fall-Through = No,
        Framed-Filter-Id=":group_name=admin"
```

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, simply add the list of group_names after the existing entries, including the separating colon ":".

9.3 Secure Management Console Access

Selecting **HTTPS Server in System: Services** enables the Administrator to establish a secure browser connection Management Console:



Service Access

Services	Network Interface	Dialout/Cellular	Dial-in	VPN
HTTP Web Management	☐	☐	☐	☐
HTTPS Web Management	☑	☑	☑	☑
Telnet command shell	☐	☐	☐	☐
SSH command shell	☑	☑	☑	☑

- Activate your preferred browser and enter https:// *IP address*. For example, if the Console Server has been set up with an IP address of 200.122.0.12, you need to type https:// 200.122.0.12 in your address bar
- 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.
- You will then be prompted for the Administrator account and password as normal.

When you have a secure HTTPS connection in place, the SSL secured icon will appear at the bottom of the browser screen. You can verify the level of encryption in place by clicking on this icon.

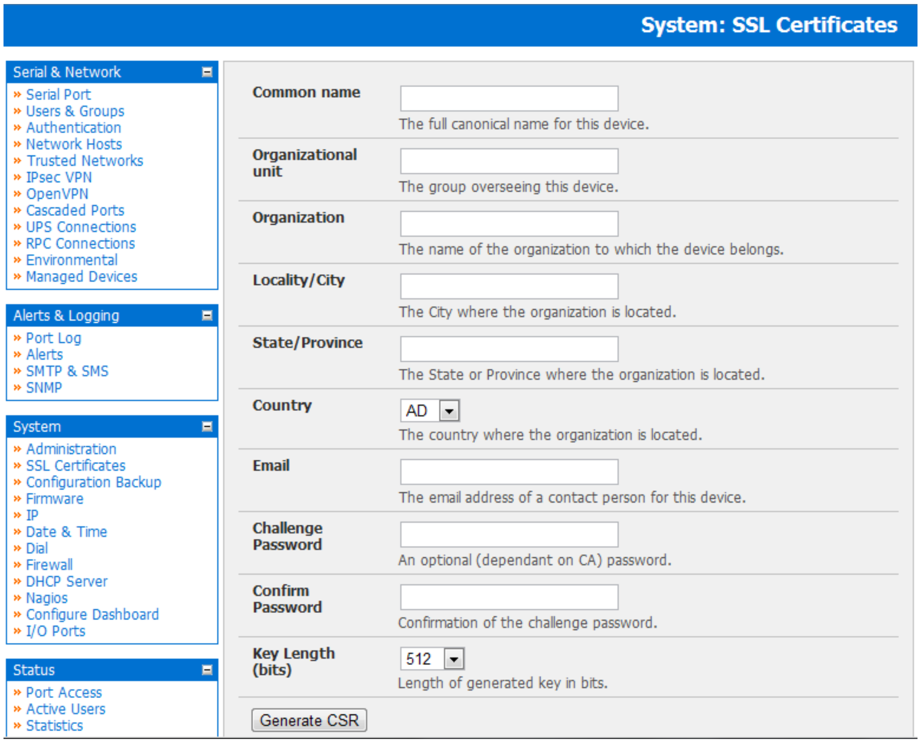
When you first enable and connect via HTTPS, it is normal that you may receive a certificate warning. The default SSL certificate in your Console Server is embedded during testing and is not signed by a recognized third party certificate authority. Rather, it is signed by our own signing authority. These warnings do not affect the encryption protection you have against eavesdroppers.

9.4 SSL Certificate

The Console Server uses the Secure Socket Layer (SSL) protocol for encrypted network traffic between itself and a connected user. During the 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 purpose only and should not be relied on for secured global access.

⚠ ***The System*** *Administrator* ***should not rely on the default certificate as the secured global access mechanism for use through Internet***

It is recommended you generate and install a new base64 X.509 certificate that is unique for a particular Console Server.



System: SSL Certificates

Common name	The full canonical name for this device.
Organizational unit	The group overseeing this device.
Organization	The name of the organization to which the device belongs.
Locality/City	The City where the organization is located.
State/Province	The State or Province where the organization is located.
Country	AD The country where the organization is located.
Email	The email address of a contact person for this device.
Challenge Password	An optional (dependant on CA) password.
Confirm Password	Confirmation of the challenge password.
Key Length (bits)	512 Length of generated key in bits.

To do this 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:

- Select **System: SSL Certificate** and fill out the fields as explained below:

Common name This is 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 that is used to access the Console Server with a web browser (without the "http://" prefix). In case 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 country where the organization is located. This is the two-letter ISO code, e.g. DE for Germany, or US for the USA. (Note: the country code has to be entered in CAPITAL LETTERS)

Email The email address of a contact person that is responsible for the Console Server and its security

Challenge Password Some certification authorities require a challenge password to authorize later changes on the certificate (e.g. revocation of the certificate). The minimal length of this password is 4 characters

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

- Once this is done, click on the button Generate CSR which will initiate the Certificate Signing Request generation. The CSR can be downloaded to your administration machine with the Download button
- 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 to the Console S*erver using the Upload button as shown below*

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 15 - Advanced*

Chapter 10: 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.

Tripp Lite Console Servers can operate in conjunction with a central/upstream Nagios server to provide distributing monitoring of attached network hosts and serial devices. The Console Servers can 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 embed a basic set of distributed monitoring add-ons and can be uploaded with additional customizable distributed monitoring.

***Note**: If you have an existing Nagios deployment, you may wish to use the Console Server in a distributed monitoring server capacity only. In this case and if you are already familiar with Nagios, skip ahead to section 10.3.*

10.1 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 are available at: http://www.nagios.org

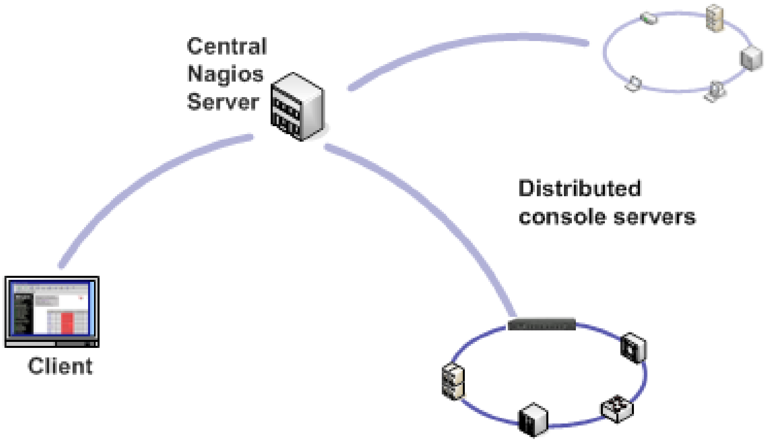
Nagios forms the core of many leading commercial system management solutions such as GroundWork: http://www.groundworkopensource.com

Nagios takes some time to install and configure, but 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, e.g., don't just check if 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

10.2 Central management and setting up SDT for Nagios

The Nagios solution has three parts: the Central Nagios server, Distributed Console Servers and the SDT for Nagios software.



Central Nagios server

- A vanilla Nagios 2.x or 3.x installation (typically on a Linux server)
- Generally running on a blade, PC, virtual machine, etc. at a central location
- Runs a web server that displays the Nagios GUI
- Imports configuration from distributed Console Servers

Distributed Console Servers

- B096-016 / B096-032 / B096-048 or B092-016 Console Servers
- Serial and network hosts are attached to each Console Server
- Each runs Nagios plug-ins, NRPE and NSCA add-ons, but not a full Nagios server

Clients

- Typically a client PC, laptop, etc. running Windows, Linux or Mac OS X
- Runs Tripp Lite SDT Connector client software 1.5.0 or later
- Connect to the central Nagios server web UI to view status of monitored hosts and serial devices
- Then use SDT Connector to connect through the distributed Console Servers, to manage monitored hosts and serial devices

10.2.1 Set up central Nagios server

The Nagios server software is available for most major distributors of Linux using the standard package management tools. Your distributor will have documentation available on how to install Nagios. This is usually the quickest and simplest way to get up and running.

Note that you will need the core Nagios server package, and at least one of the NRPE or NSCA add-ons. NSCA is required to utilize the alerting features of the distributed hosts; installing both NRPE and NSCA is recommended.

You will also require a web server such as Apache to display the Nagios web UI (and this may be installed automatically as a dependency of the Nagios packages).

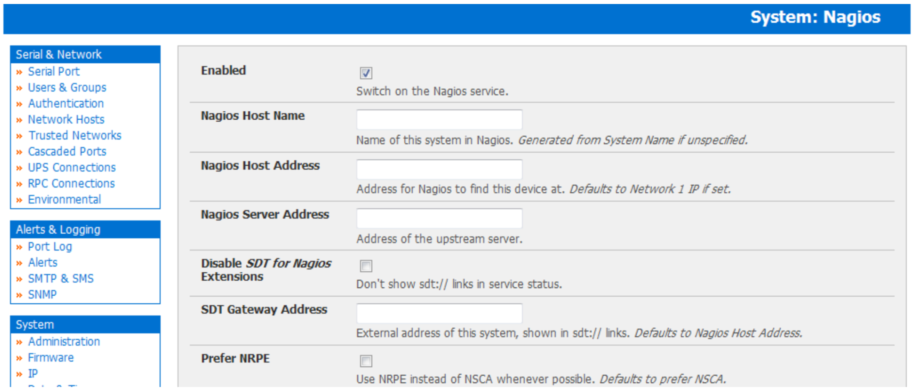
Alternatively, you may wish to download the Nagios source code directly from the Nagios website, and build and install the software from scratch. The Nagios website (http://www.nagios.org) has several Quick Start Guides that walk through this process.

Once you are able to browse to your Nagios server and see its web UI and the local services it monitors by default, you are ready to continue.

10.2.2 Set up distributed Console Servers

This section provides a brief walk-through on configuring a single Console Server to monitor the status of one attached network host (a Windows IIS server running HTTP and HTTPS services) and one serially attached device (the console port of a network router), and to send alerts back to the Nagios server when an administrator connects to the router or IIS server.

While this walk-through provides an example, details of the configuration options are described in the next section. This walk-through also assumes the network host and serial devices are already physically connected to the Console Server. First step is to set up the Nagios features on the Console Server:



Enabled ☑ Switch on the Nagios service.

Nagios Host Name Name of this system in Nagios. *Generated from System Name if unspecified.*

Nagios Host Address Address for Nagios to find this device at. *Defaults to Network 1 IP if set.*

Nagios Server Address Address of the upstream server.

Disable *SDT for Nagios* Extensions □ Don't show sdt:// links in service status.

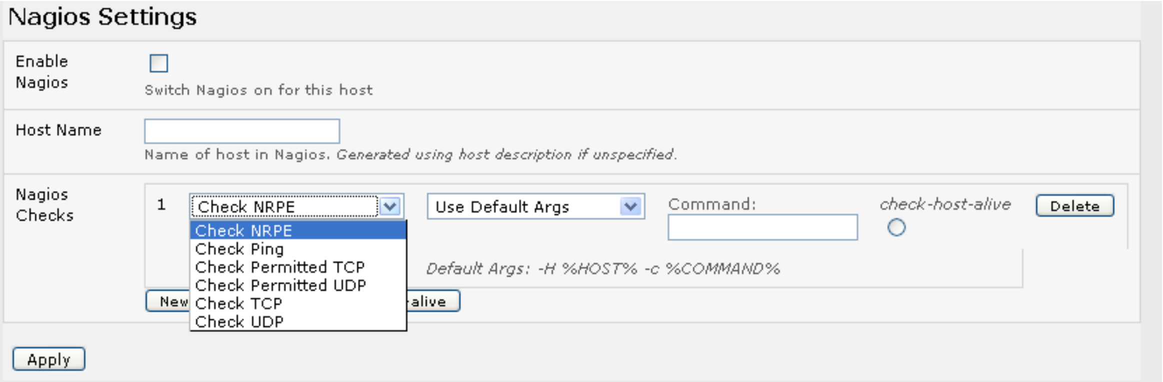
SDT Gateway Address External address of this system, shown in sdt:// links. *Defaults to Nagios Host Address.*

Prefer NRPE □ Use NRPE instead of NSCA whenever possible. *Defaults to prefer NSCA.*

- Browse the Console Server and select **System: Nagios** on the Console Server Management Console. Check Nagios service **Enabled**
- Enter the **Host Name** and the **Nagios Host Address** (i.e. IP address) that the central Nagios server will use to contact the distributed Console Server
- Enter the IP address that the distributed Console Server will use to contact the central Nagios server in **Nagios Server Address**
- Enter the IP address that the clients running SDT Connector will use to connect through the distributed Console Servers in **SDT Gateway address**
- Check **Prefer NRPE, NRPE Enabled** and **NRPE Command Arguments**
- Check **NSCA Enabled**, choose an **NSCA Encryption Method** and enter and confirm an **NSCA Secret**. Remember these details as you will need them later on. For **NSCA Interval**, enter *5*
- Click **Apply**.

Next, configure the attached Window network host and specify the services you will be checking with Nagios (HTTP and HTTPS):

- Select **Network Hosts** from the **Serial & Network** menu and click **Add Host**.
- Enter the **IP Address/DNS Name** of the network server, (e.g.: *192.168.1.10*) and enter a **Description**, (e.g.: *Windows 2003 IIS Server*)
- Remove all **Permitted Services.** This server will be accessible using Terminal Services, so check **TCP**, **Port 3389** and log **level 1** and click **Add.** It is important to remove and re-add the service to enable logging



Enable Nagios □ Switch Nagios on for this host

Host Name — Name of host in Nagios. *Generated using host description if unspecified.*

Nagios Checks 1 Check NRPE — Use Default Args — Command: — *check-host-alive* ○ — *Default Args: -H %HOST% -c %COMMAND%*

- Scroll down to **Nagios Settings** and check **Enable Nagios**
- Click **New Check** and select **Check Ping**. Click **check-host-alive**
- Click **New Check** and select **Check Permitted TCP**. Select **Port 3389**
- Click **New Check** and select **Check TCP**. Select **Port 80**
- Click **New Check** and select **Check TCP**. Select **Port 443**
- Click **Apply**

Similarly, configure the serial port to the router to be monitored by Nagios:

- Select **Serial Port** from the **Serial & Network** menu
- Locate the serial port that has the router console port attached and click **Edit**
- Ensure the serial port settings under *Common Settings* are correct and match the attached router's console port
- Click **Console Server Mode** and select **Logging Level 1**
- Check **Telnet** (SSH access is not required, as SDT Connector is used to secure the otherwise unsecured Telnet connection)
- Scroll down to **Nagios Settings** and check **Enable Nagios**
- Check **Port Log** and **Serial Status**
- Click **Apply**

Now set the Console Server to send alerts to the Nagios server

- Select **Alerts** from the **Alerts & Logging** menu and click **Add Alert**
- In **Description** enter: *Administrator* connection
- Check **Nagios (NSCA)**
- In **Applicable Ports** check the serial port that has the router console port attached. In **Applicable Hosts** check the IP address/DNS name of the IIS server
- Click **Connection Alert**
- Click **Apply**

Lastly, add a User for the client running SDT Connector:

- Select *Users & Groups* from the *Serial & Network* menu
- Click **Add User**
- In **Username**, enter: *sdtnagiosuser*, then enter and confirm a **Password**
- In **Accessible Hosts** click the IP address/DNS name of the IIS server. In **Accessible Ports** click the serial port that has the router console port attached
- Click **Apply**

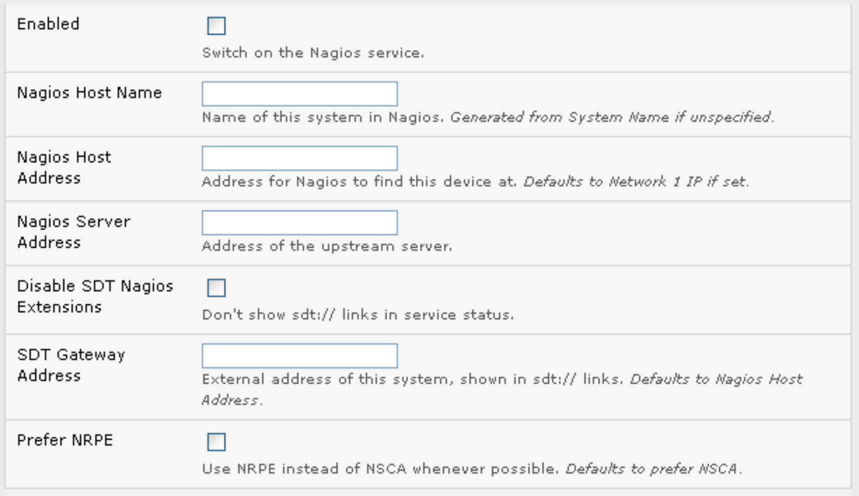
10.3 Configuring Nagios distributed monitoring

To activate the Console Server's 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 which are to be monitored must have Nagios enabled and any specific Nagios checks configured
- Lastly the central/upstream Nagios monitoring host must be configured

10.3.1 Enable Nagios on the Console Server

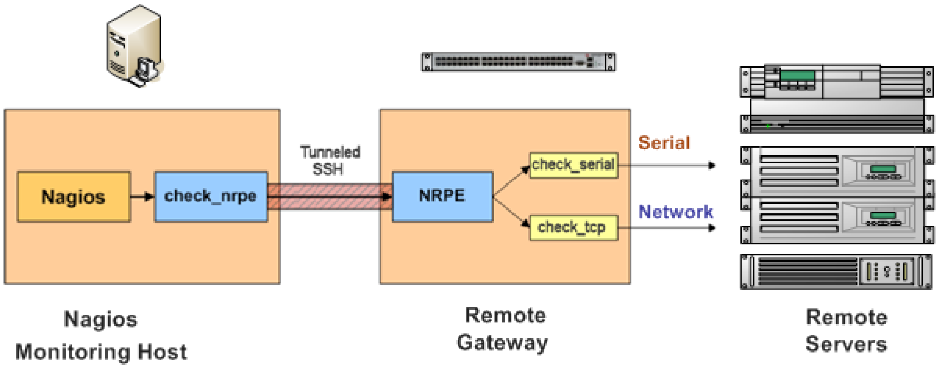
- Select **System: Nagios** on the Console Server Management Console and tick the Nagios service **Enabled**



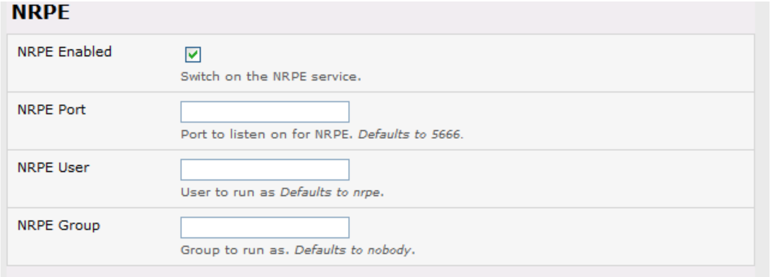
Enabled	□ Switch on the Nagios service.
Nagios Host Name	 Name of this system in Nagios. *Generated from System Name if unspecified.*
Nagios Host Address	 Address for Nagios to find this device at. *Defaults to Network 1 IP if set.*
Nagios Server Address	 Address of the upstream server.
Disable SDT Nagios Extensions	□ Don't show sdt:// links in service status.
SDT Gateway Address	 External address of this system, shown in sdt:// links. *Defaults to Nagios Host Address.*
Prefer NRPE	□ Use NRPE instead of NSCA whenever possible. *Defaults to prefer NSCA.*

- Enter the **Nagios Host Name** that the Console Server will be referred to in the Nagios central server – this will be generated from local System Name (entered in **System: Administration**) if unspecified
- In **Nagios Host Address**, enter the IP address or DNS name that the upstream Nagios server will use to reach the Console Server – if unspecified this will default to the first network port's IP as entered in **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 if you wish to disable the SDT Connector integration with your Nagios server at the head end – this would only be checked if you want to run a 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**
- When NRPE and NSCA are both enabled, NSCA is preferred method for communicating with the upstream Nagios server – check **Prefer NRPE** to use NRPE whenever possible (i.e. for all communication except for alerts)

10.3.2 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 which is especially valuable if you are monitoring hundreds or thousands of hosts. To enable NRPE:



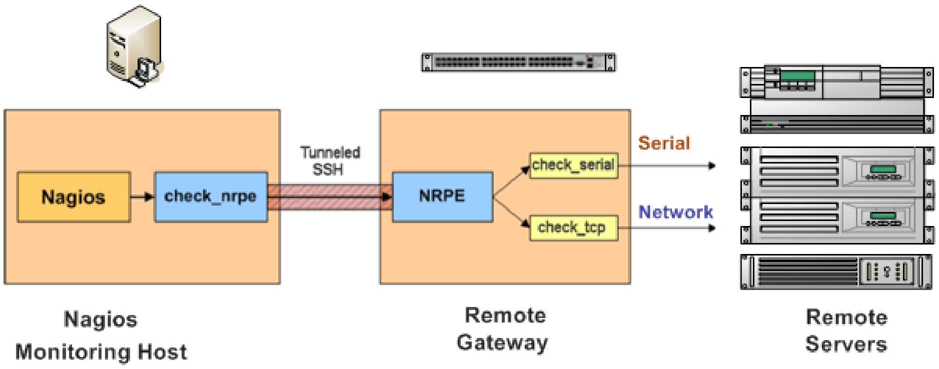
NRPE

NRPE Enabled	☑ Switch on the NRPE service.
NRPE Port	 Port to listen on for NRPE. *Defaults to 5666.*
NRPE User	 User to run as *Defaults to nrpe.*
NRPE Group	 Group to run as. *Defaults to nobody.*

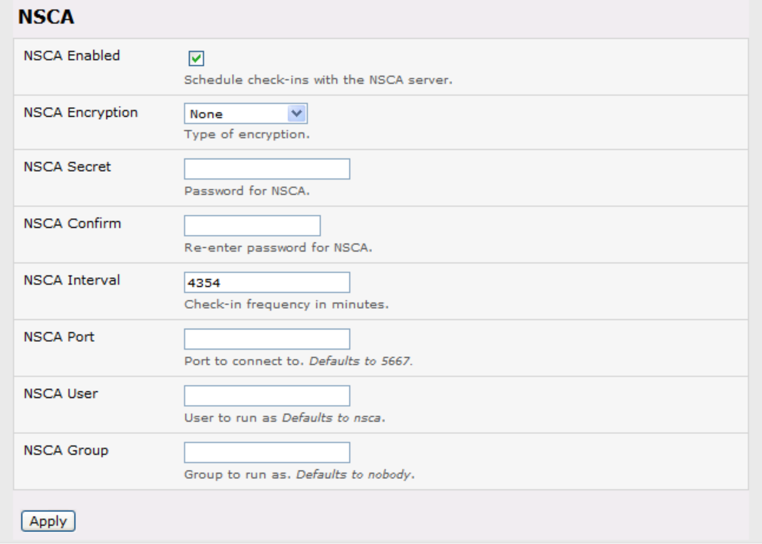
- Select **System: Nagios** and check **NRPE Enabled**
- Enter the details for the user connection to the upstream Nagios monitoring server. Again, refer to the sample Nagios configuration example below for details of configuring specific NRPE checks

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.

10.3.3 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. To enable NSCA:



NSCA

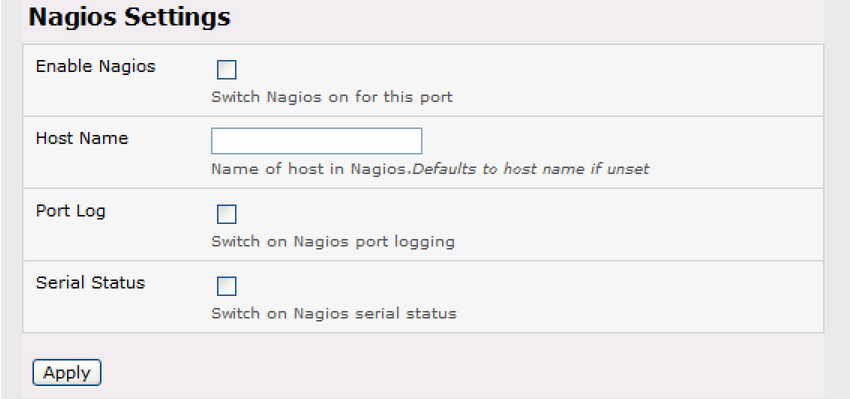
NSCA Enabled	☑ Schedule check-ins with the NSCA server.
NSCA Encryption	None Type of encryption.
NSCA Secret	 Password for NSCA.
NSCA Confirm	 Re-enter password for NSCA.
NSCA Interval	4354 Check-in frequency in minutes.
NSCA Port	 Port to connect to. *Defaults to 5667.*
NSCA User	 User to run as *Defaults to nsca.*
NSCA Group	 Group to run as. *Defaults to nobody.*

- Select **System: Nagios** and check **NSCA Enabled**
- Select the **Encryption** to be used from the drop-down menu, then enter a **Secret** password and specify a check **Interval**
- Refer the sample Nagios configuration section below for some examples of configuring specific NSCA checks

10.3.4 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. Refer to *Chapter 4.4: Network Host Configuration* for details on enabling Nagios monitoring for Hosts that are network connected to the Console Server. To enable Nagios to monitor a device connected to the Console Server serial port:

- Select **Serial & Network: Serial Port** and click **Edit** on the serial Port # to be monitored
- Select **Enable Nagios**, specify the name of the device on the upstream server and determine the check to be run on this port. **Serial Status** monitors the handshaking lines on the serial port and **Check Port** monitors the data logged for the serial port



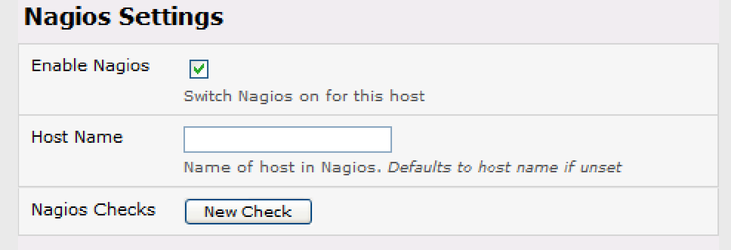
Nagios Settings

Enable Nagios	□ Switch Nagios on for this port
Host Name	 Name of host in Nagios.*Defaults to host name if unset*
Port Log	□ Switch on Nagios port logging
Serial Status	□ Switch on Nagios serial status

10.3.5 Configure selected Network Hosts for Nagios monitoring

The individual Network Hosts connected to the Console Server that is to be monitored must also be configured for Nagios checks:

- Select **Serial & Network: Network Port** and click **Edit** on the Network Host to be monitored



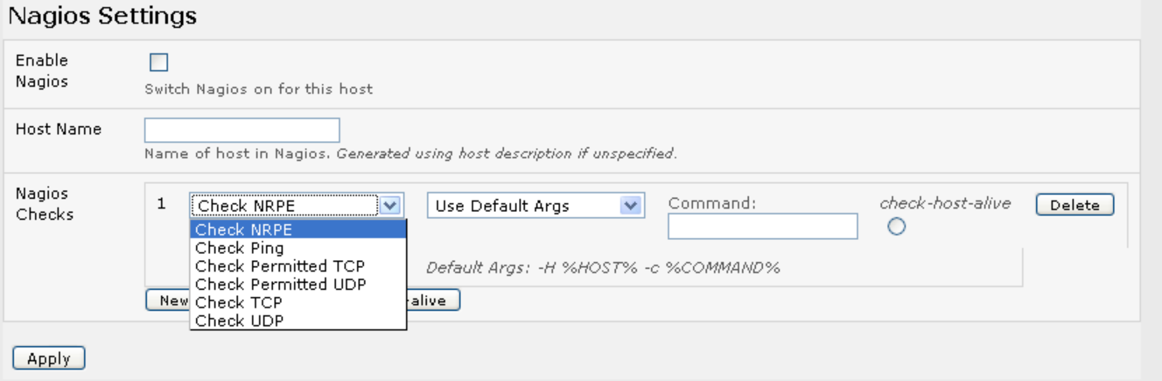
Nagios Settings

Enable Nagios ☑ Switch Nagios on for this host

Host Name [ ] Name of host in Nagios. *Defaults to host name if unset*

Nagios Checks [New Check]

- Select **Enable Nagios**, specify the 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/UDP** to monitor a service that you have previously added as a **Permitted Service**
- Select **Check TCP/UDP** to specify a service port that you wish to monitor, but do not wish to allow external (SDT Connector) access
- Select **Check TCP** to monitor



Nagios Settings

Enable Nagios □ Switch Nagios on for this host

Host Name [ ] Name of host in Nagios. *Generated using host description if unspecified.*

Nagios Checks 1 [Check NRPE ▾] [Use Default Args ▾] Command: [ ] *check-host-alive* ○ [Delete]

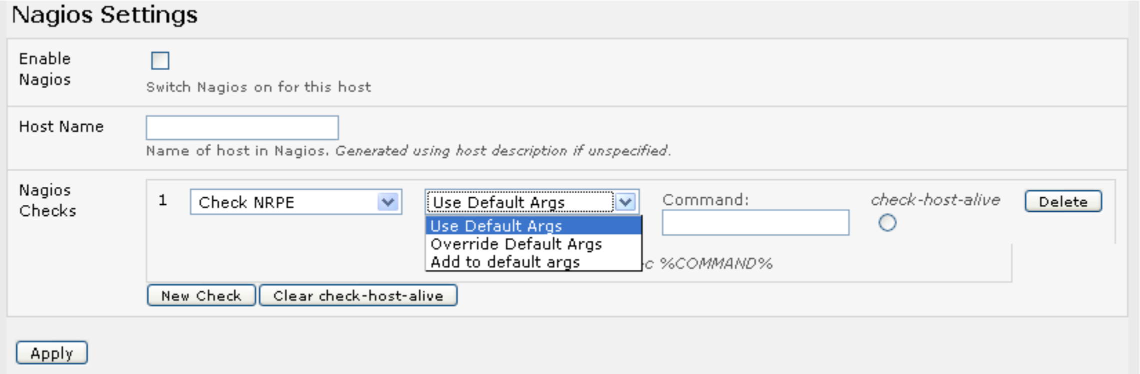
Check NRPE
Check Ping
Check Permitted TCP
Check Permitted UDP
Check TCP
Check UDP

Default Args: -H %HOST% -c %COMMAND%

[New …] […alive]

[Apply]

- The **Nagios Check** nominated as the ***check-host-alive*** check is 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
- If no ***check-host-alive*** check is selected, the host will always be assumed to be up
- You may deselect ***check-host-alive*** by clicking **Clear *check-host-alive***
- If required, customize the selected **Nagios Checks** to use custom arguments
- Click **Apply**



Nagios Settings

Enable Nagios □ Switch Nagios on for this host

Host Name [ ] Name of host in Nagios. *Generated using host description if unspecified.*

Nagios Checks 1 [Check NRPE ▾] [Use Default Args ▾] Command: [ ] *check-host-alive* ○ [Delete]

Use Default Args
Override Default Args
Add to default args

…c %COMMAND%

[New Check] [Clear check-host-alive]

[Apply]

10.3.6 Configure the upstream Nagios monitoring host

Refer to the Nagios documentation (http://www.nagios.org/docs/) for configuring the upstream server:

- The section entitled *Distributed Monitoring* steps through what is needed to configure NSCA on the upstream server (under *Central Server Configuration*)
- *NRPE Documentation*, which has been recently added, 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 is configured, and individual serial port and network host connections on the Console Server are 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.

10.4 Advanced Distributed Monitoring Configuration

10.4.1 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. Each check has two configurations, one for NRPE and one for NSCA. In practice, these would be combined into a single check which uses NSCA as a primary method and falling back to NRPE if a check were late. For details, see the Nagios documentation (http://www.nagios.org/docs/) on *Service and Host Freshness Checks*.

```
; Host definitions
;
; Console Server
define host{
        use                  generic-host
        host_name                 tripplite
        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_namecheck_nrpe_daemon
            command_line    $USER1$/check_nrpe -H 192.168.254.147 -p 5666
            }

define service {
            service_description          NRPE Daemon
            host_name                    tripplite
            use                          generic-service
            check_command                            check_nrpe_daemon
            }

; Serial Status
define command {
            command_namecheck_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                               tripplite_nrpe_daemon_dep
        host_name                          tripplite
        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                               tripplite_nrpe_daemon_dep
        host_name                          tripplite
        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_tripplite
        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_tripplite
        }

define service {
        service_description       host-ping-server
        host_name                 server
        use                       generic-service
        check_command                       check_ping_via_tripplite
        active_checks_enabled     0
        passive_checks_enabled 1
        }

define servicedependency{
        name                                tripplite_nrpe_daemon_dep
        host_name                           tripplite
        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_tripplite
      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_tripplite!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_tripplite!tcp!22
        active_checks_enabled     0
        passive_checks_enabled 1
        }

define servicedependency{
        name                                tripplite_nrpe_daemon_dep
        host_name                           tripplite
        dependent_host_name                 server
        dependent_service_description       SSH Port
        service_description                 NRPE Daemon
        execution_failure_criteria w,u,c
        }
```

10.4.2 Basic Nagios plug-ins

Plug-ins are compiled executables or scripts that can be scheduled to be run on the Console Server to check the status of a connected host or service. This status is then communicated to the upstream Nagios server which uses the results to monitor the current status of the distributed network. Each Console Server is preconfigured with a selection of the checks that are part of the Nagios plug-ins package:

check_tcp and *check_udp* are used to check open ports on network hosts

check_ping is used to check network host availability

check_nrpe is used to execute arbitrary plug-ins in other devices

Each Console Server is also preconfigured with two checks that are specific to the Console Server:

check_serial_signals is used to monitor the handshaking lines on the serial ports

check_port_log is used to monitor the data logged for a serial port.

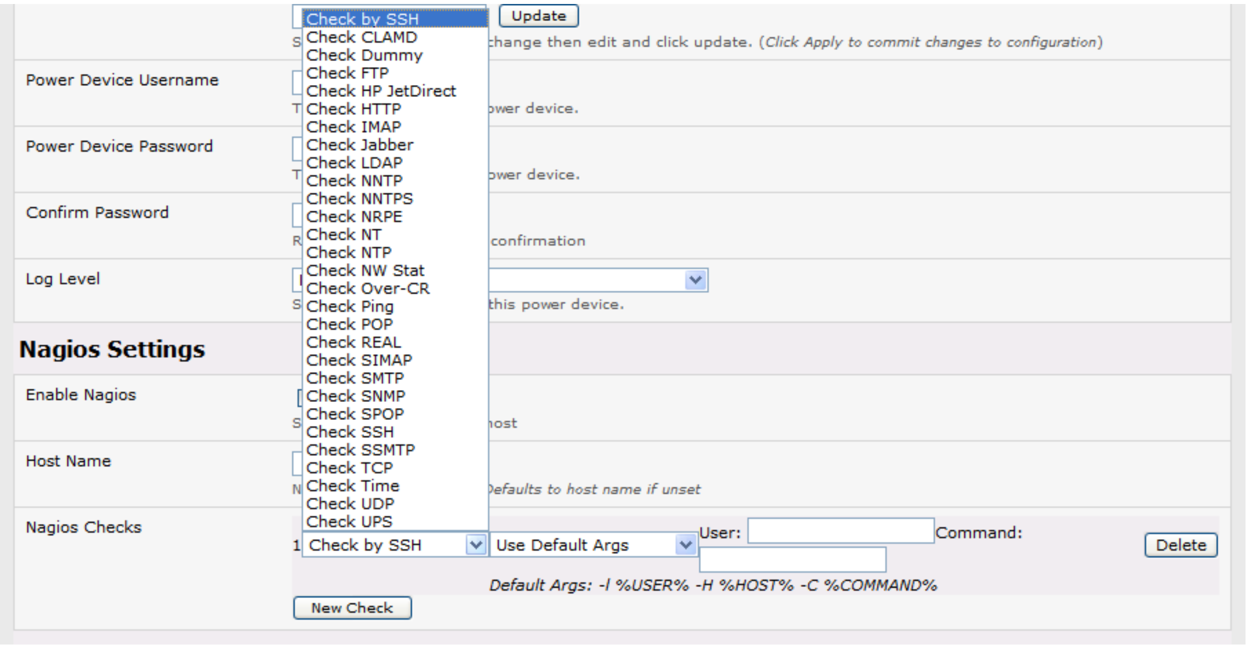
10.4.3 Additional plug-ins

Additional Nagios plug-ins (listed below) are available for all the Tripp Lite Console Servers:

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

There also are *bash* scripts which can be downloaded and run (*primarily check_log.sh*).

- To configure additional checks, the downloaded plug-in program must be saved in the tftp *addins* directory on the USB flash and the downloaded text plug-in file saved in */etc/config*
- To enable these new additional checks, you select **Serial&Network: Network Port**, then you **Edit** the Network Host to be monitored, and select **New Checks**. The additional check option will have been included in the updated **Nagios Checks** list. You can again customize the arguments



change then edit and click update. (*Click Apply to commit changes to configuration*)

Power Device Username

power device.

Power Device Password

power device.

Confirm Password

confirmation

Log Level

this power device.

Nagios Settings

Enable Nagios

host

Host Name

Defaults to host name if unset

Nagios Checks

1 Check by SSH | Use Default Args | User: | Command:

Default Args: -l %USER% -H %HOST% -C %COMMAND%

Chapter 11: System Management

This chapter describes how the Administrator can perform a range of general system administration and configuration tasks on the Console Server, such as:

- Applying *Soft* and *Hard* Resets to the gateway
- Re-flashing the firmware
- Configuring the Date, Time and NTP
- Setting up Backup of the configuration files (B095-004/003 only)
- Configuring the console server in FIPS mode(B095-004/003 only)
- Delayed configuration commits

System administration and configuration tasks covered elsewhere include:

- Resetting the System Password and entering a new System Name and Description for the Console Server (*Chapter 3.2*)
- Setting the Console Server's System IP Address (*Chapter 3. 3*)
- Setting the permitted Services used to access the Console Server (*Chapter 3.4*)
- Setting up OoB Dial-in (*Chapter 5*)

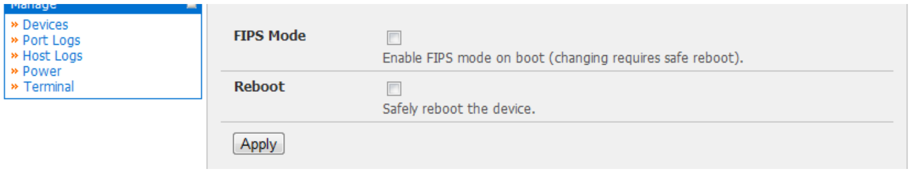
Configuring the Dashboard (B095-004/003 only) (*Chapter 12*)

11.1 System Administration and Reset

The Administrator can reboot or reset the gateway to default settings.

A *soft* reset is affected by:

- Selecting **Reboot** in the **System: Administration** menu and clicking **Apply**



FIPS Mode ☐ Enable FIPS mode on boot (changing requires safe reboot).

Reboot ☐ Safely reboot the device.

The Console Server reboots with all settings (*e.g.* the assigned network IP address) preserved. However this *soft* reset does disconnect all users and ends any SSH sessions that had been established.

A *soft* reset will also be affected when you switch OFF power from the Console Server, and then switch the power back ON. However if you cycle the power while the unit is writing to flash you could corrupt or lose data, so the software reboot is the safer option.

A *hard erase* (*hard reset*) is effected by:

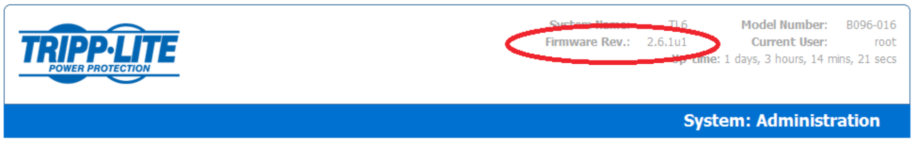
- Pushing the *Erase* button on the rear panel **twice**. A ball point pen or bent paper clip is a suitable tool for performing this procedure. Do not use a graphite pencil. Depress the button gently **twice** (within a couple of second period) while the unit is powered ON.

This will reset the Console Server back to its factory default settings and clear the Console Server's stored configuration information (*i.e.* the 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 administration password (Username: **root** Password: **default**).

11.2 Upgrade Firmware

Before upgrading you should ascertain if you are already running the most current firmware in your gateway. Your Console Server will not allow you to upgrade to the same or an earlier version.

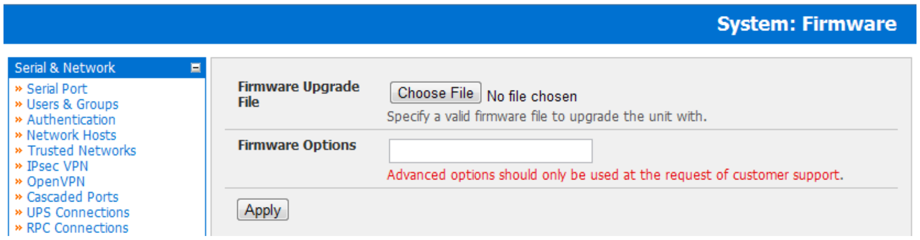
- The **Firmware** version is displayed in the header of each page



System Name: TL6 Model Number: B096-016
Firmware Rev.: 2.6.1u1 Current User: root
Up time: 1 days, 3 hours, 14 mins, 21 secs

System: Administration

- Or select **Status: Support Report** and note the **Firmware Version**
- To upgrade, you must first download the latest firmware image from http://www.tripplite.com/EN/support/downloads/driver-firmware-downloads.cfm
- Save this downloaded firmware image file on to a system on the same subnet as the Console Server
- Also download and read the *release_notes.txt* for the latest information
- To then upload the firmware image file to your Console Server, select **System: Firmware**



System: Firmware

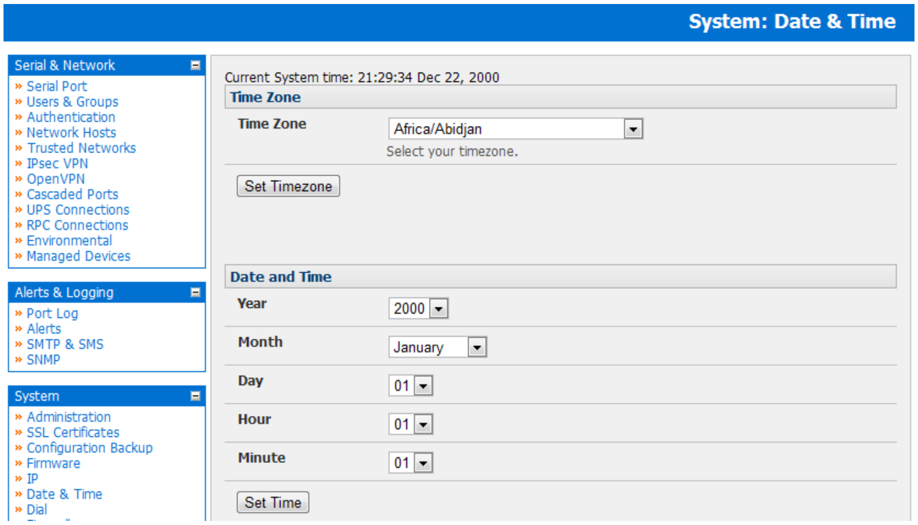
Firmware Upgrade File: No file chosen
Specify a valid firmware file to upgrade the unit with.

Firmware Options:
Advanced options should only be used at the request of customer support.

- Specify the address and name of the downloaded Firmware Upgrade File, or **Browse** the local subnet and locate the downloaded file
- Click **Apply** and the Console Server appliance 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 Console Server will have retained all its pre-upgrade configuration information

11.3 Configure Date and Time

It is recommended that you set the local Date and Time in the Console Server 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.



Current System time: 21:29:34 Dec 22, 2000

Time Zone

Time Zone: Africa/Abidjan
Select your timezone.

Date and Time

Year: 2000
Month: January
Day: 01
Hour: 01
Minute: 01

- Select the **System: Date & Time** menu option
- Manually set the **Year, Month, Day, Hour** and **Minute** using the **Date** and **Time** selection boxes, then click **Set Time**

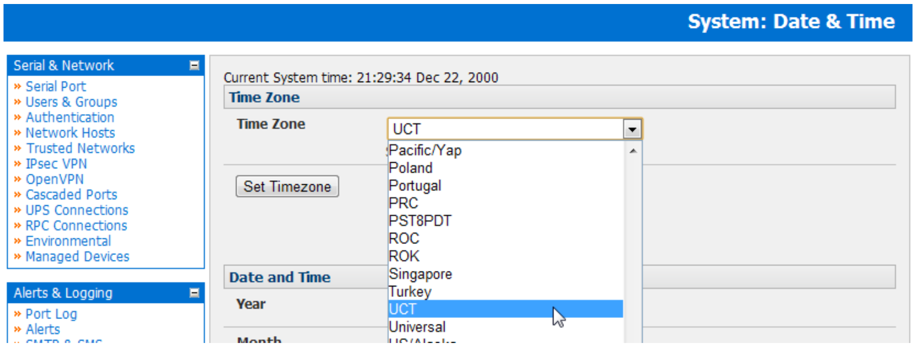
The gateway can synchronize its system time with a remote time server using the Network Time Protocol (NTP). Configuring the NTP time server ensures that the Console Server clock will be accurate soon after the Internet connection is established. Also if NTP is not used, the system clock will be reset randomly every time the Console Server is powered up. To set the system time using NTP:

- Select the **Enable NTP** checkbox on the **Network Time Protocol** page
- Enter the IP address of the remote **NTP Server**
- If your external NTP server requires authentication, you need to specify the **NTP Authentication Key** and the **Key Index** to use when authenticating with the NTP server
- Click **Apply Settings**

You must now also specify your local time zone so the system clock can show local time (and not UTP):

- Set your appropriate region/locality in the Time Zone selection box and click **Set Timezone**

The **Time Zone** can also be set to **UCT (Coordinated Universal Time)** which replaced Greenwich Mean Time as the World standard for time in 1986:



Current System time: 21:29:34 Dec 22, 2000

Time Zone

Time Zone: UCT

Date and Time

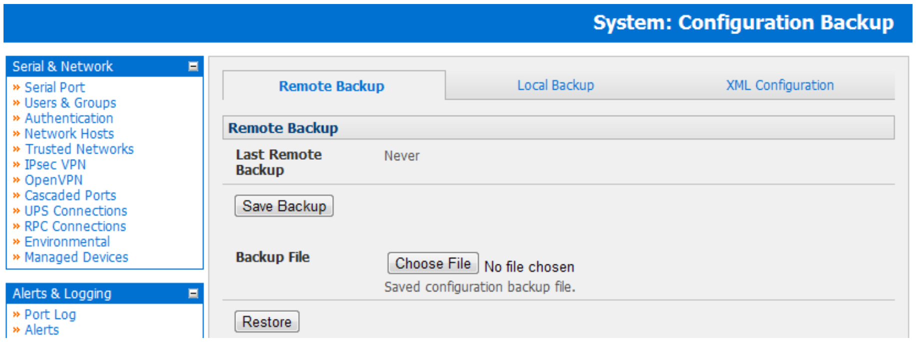
Year

11.4 Configuration Backup

It is recommended 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 the System: Configuration Backup menu option or click the Backup icon

***Note**: The configuration files can also be backed up from the command line (refer **Chapter 14**)*



Remote Backup

Last Remote Backup Never

Backup File No file chosen
Saved configuration backup file.

You can save the backup file remotely on your PC and you can restore configurations from remote locations:

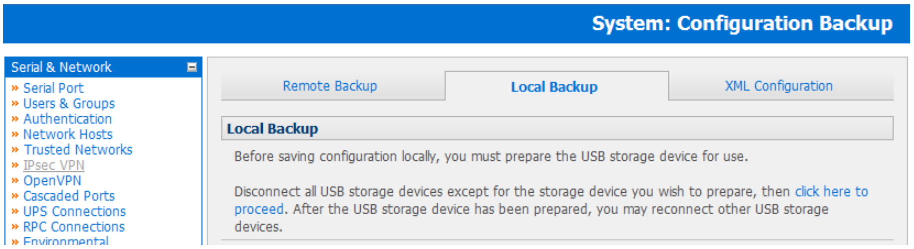
- Click **Save Backup** in the Remote Configuration Backup menu
- The config backup file (*System Name_date_config.opg*) will be downloaded to your PC and saved in the location you nominate

To restore a remote backup:

- Click **Browse** in the Remote Configuration Backup menu and select the **Backup File** you wish to restore
- Click **Restore** and click **OK**. This will overwrite all the current configuration settings in your Console Server

Alternately 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 is the only USB device attached to the Console Server
- Select the **Local Backup** tab and **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. However it is recommended 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.



Local Backup

Before saving configuration locally, you must prepare the USB storage device for use.

Disconnect all USB storage devices except for the storage device you wish to prepare, then click here to proceed. After the USB storage device has been prepared, you may reconnect other USB storage devices.

- To backup 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
- To restore a backup from the USB simply 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** and click **Apply**

***Note**: Before selecting **Load On Erase** please ensure you have tested your alternate default configuration by clicking Restore*

If for some reason your alternate default configuration causes the Console Server to become unbootable recover your unit to factory settings using the following steps:

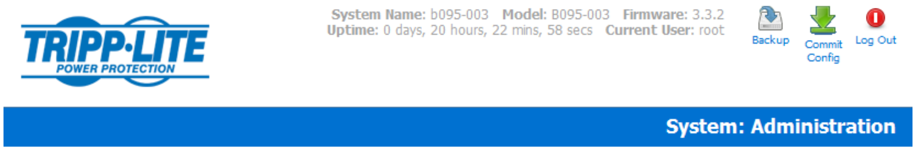
- *If the configuration is stored on an external USB storage device, unplug the storage device and reset to factory defaults as per section 11.1 of the user manual*
- *If the configuration is stored on an internal USB storage device reset to factory defaults using a specially prepared USB storage device:*
    - o *The 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 already formatted this way*
    - o *The file system must have the volume label: OPG_DEFAULT*
    - o *Insert this USB storage device into an external USB port on the Console Server and reset to factory defaults as per section 11.1*

After recovering your Console Server, ensure the problematic configuration is no longer selected for Load On Erase

11.5 Delayed Configuration Commit

The Delayed Config Commit 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:

- Check the **Delayed Config Commits** button under **System: Administration**
- Click **Apply**
- The Commit Config icon Commit Config will now be displayed in top right-hand corner of the screen between the Backup and Log Out icons

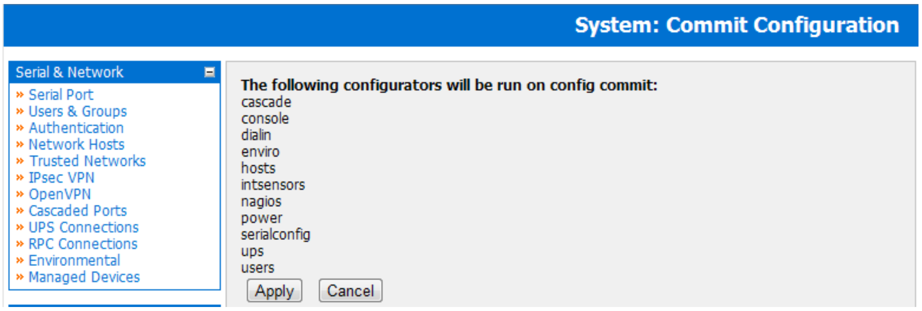


System Name: b095-003   Model: B095-003   Firmware: 3.3.2
Uptime: 0 days, 20 hours, 22 mins, 58 secs   Current User: root

System: Administration

To queue then run configuration changes:

- Firstly apply all the required changes to the configuration e.g. 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



System: Commit Configuration

The following configurators will be run on config commit:

```
cascade
console
dialin
enviro
hosts
intsensors
nagios
power
serialconfig
ups
users
```

- Click **Apply** to run all the configurators in the queue
- Alternately click **Cancel** and this will discard all the delayd configuration changes

Note*: All the queued configuration changes will be lost if Cancel is selected*
To disable the Delayed Configuration Commits mode:

- Uncheck the **Delayed Config Commits** button under **System: Administration** and click **Apply**
- Click the **Commit Config** button in top right-hand corner of the screen to display the **System: Commit Configuration** screen
- Click **Apply** to run the systemsettings configurator

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.

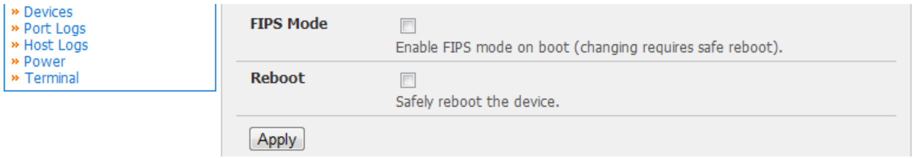
11.6 FIPS Mode

Note*: The US National Institute of Standards and Technology (NIST) publishes the FIPS (Federal Information Processing Standard) series of standards. FIPS 140-1 and FIPS 140-2 are both technical standards and worldwide de-facto standards for the implementation of cryptographic modules. These standards and guidelines are issued by NIST for use government-wide. NIST develops FIPS when there are compelling Federal government requirements such as for security and interoperability and there are no acceptable industry standards or solutions.*

Console Servers with Revision 3.0.1 firmware (or later) use an embedded OpenSSL cryptographic module that has been validated to meet the FIPS 140-2 standards and has received Certificate #1051. This firmware is only currently available on B095-004-1E / B095-003-1E-M Console Servers

When configured in FIPs mode all SSH, HTTPS and SDTConnector access to all services on the Console Servers will use the embedded FIPS compliant cryptographic module. To connect you must also be using cryptographic algorithms that are FIPs approved in your browser or client or the connection will fail.

- Select the **System: Administration** menu option
- Check **FIPS Mode** to enable FIPS mode on boot, and check **Reboot** to safely reboot the console server



FIPS Mode □ Enable FIPS mode on boot (changing requires safe reboot).

Reboot □ Safely reboot the device.

- Click **Apply** and the Console Server will now reboot. It will take several minutes to reconnect as secure communications with your browser are validated, and when reconnected it will display "*FIPs mode: Enabled*" in the banner

Note*: 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:

```
config -d config.system.fips
rm /etc/config/FIPS
flatfsd –b
```

Chapter 12: Status Reports

This chapter describes the dashboard feature and the status reports that are available:

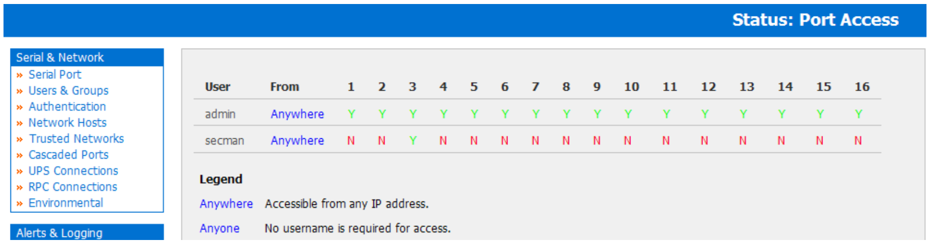
- Port Access and Active Users
- Statistics
- Support Reports
- Syslog
- Dashboard

The UPS, RPC and Environmental Status reports are covered in Chapter 8

12.1 Port Access and Active Users

The Administrator can see which Users have access privileges with which serial ports:

- Select the **Status: Port Access**



User	From	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
admin	Anywhere	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
secman	Anywhere	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N

Legend

Anywhere	Accessible from any IP address.
Anyone	No username is required for access.

The Administrator can also see the current status as to Users who have active sessions on those ports:

- Select the **Status: Active Users**

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, an RFC2217 session cannot be disconnected.

The *root* user (or any user in the *admin* group) can access the **Active Users** page. The **Active Users** page shows a snapshot of the connected sessions 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 side of the table that list active users. These buttons disconnect all sessions from their corresponding *Port*. 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 proceed to connect and disconnect.

After pressing the buttons, the selected sessions will be disconnected and the number of disconnect sessions will be displayed to the user.

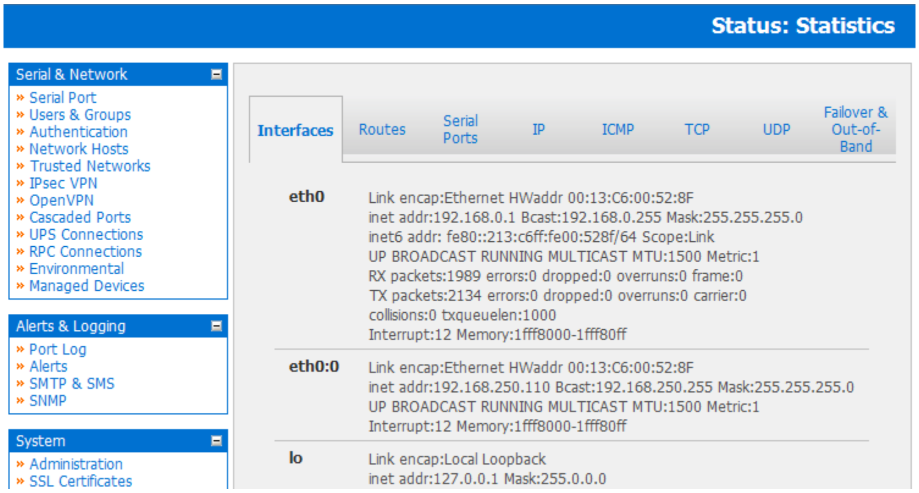
To allow more detailed control of whom to disconnect, a table with drop down lists is located at the bottom of the page for all connected users and all connected ports that allow the user to choose from whom to disconnect. For example, if you wish to disconnect the user *'tester'* from all ports, choose *'tester'* in the *Users* box, and *All ports* in the *Ports* box. Then click the *Disconnect Sessions* button.

***Note:** You can also disconnect serial sessions from the command line using the* --disconnect *option with the* pmusers *command.*

12.2 Statistics

The Statistics report provides a snapshot of the status, current traffic and other activities and operations of your Console Server:

- Select the **Status: Statistics**

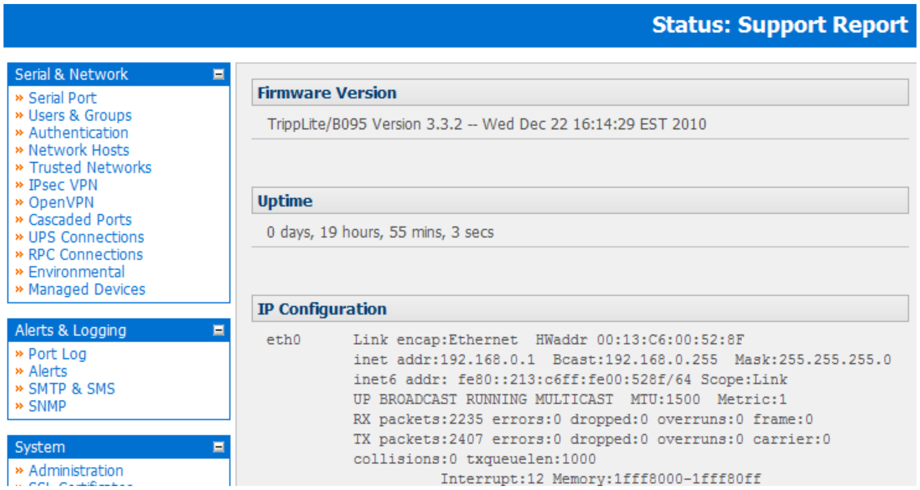


```
eth0    Link encap:Ethernet HWaddr 00:13:C6:00:52:8F
        inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0
        inet6 addr: fe80::213:c6ff:fe00:528f/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:1989 errors:0 dropped:0 overruns:0 frame:0
        TX packets:2134 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        Interrupt:12 Memory:1fff8000-1fff80ff

eth0:0  Link encap:Ethernet HWaddr 00:13:C6:00:52:8F
        inet addr:192.168.250.110 Bcast:192.168.250.255 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        Interrupt:12 Memory:1fff8000-1fff80ff

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
```

12.3 Support Reports

The Support Report provides useful status information that will assist the Tripp Lite technical support team to resolve any issues you may experience with your Console Server.

If you do experience an issue and have to contact Support, ensure you include the Support Report with your email support request. The Support Report should be generated when the issue is occurring, and attached in plain text format.



Firmware Version

TrippLite/B095 Version 3.3.2 -- Wed Dec 22 16:14:29 EST 2010

Uptime

0 days, 19 hours, 55 mins, 3 secs

IP Configuration

```
eth0      Link encap:Ethernet  HWaddr 00:13:C6:00:52:8F
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::213:c6ff:fe00:528f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2235 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2407 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
                    Interrupt:12 Memory:1fff8000-1fff80ff
```

- Select **Status: Support Report** and you will be presented with a status snapshot
- Save the file as a text file and attach it to your support email

12.4 Syslog

The Linux System Logger in the Console Server maintains a record of all system messages and errors:

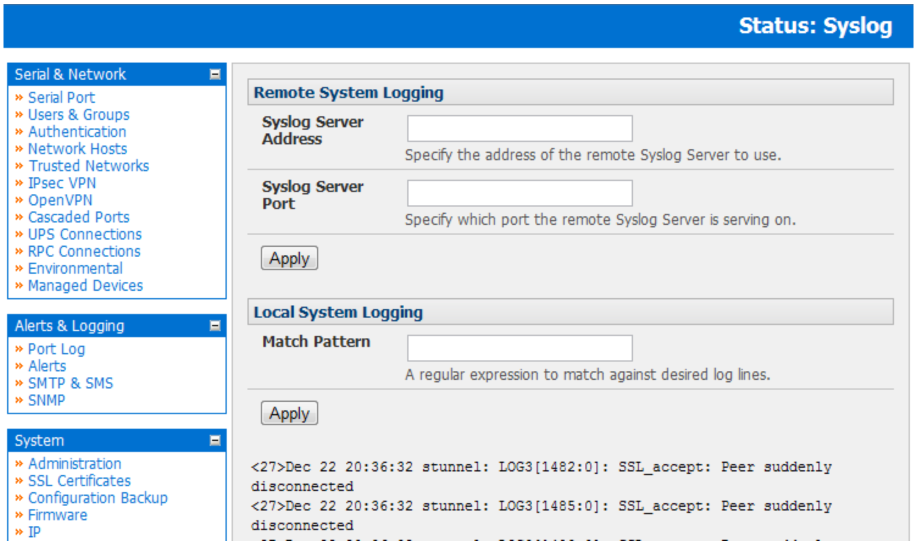
- Select **Status: Syslog**

The syslog record can be redirected to a remote Syslog Server:

- Enter the remote **Syslog Server Address** and **Syslog Server Port** details and click **Apply**

The console maintains a local Syslog. To view the local Syslog file:

- Select **Status: Syslog**



Remote System Logging

Syslog Server Address

Specify the address of the remote Syslog Server to use.

Syslog Server Port

Specify which port the remote Syslog Server is serving on.

Local System Logging

Match Pattern

A regular expression to match against desired log lines.

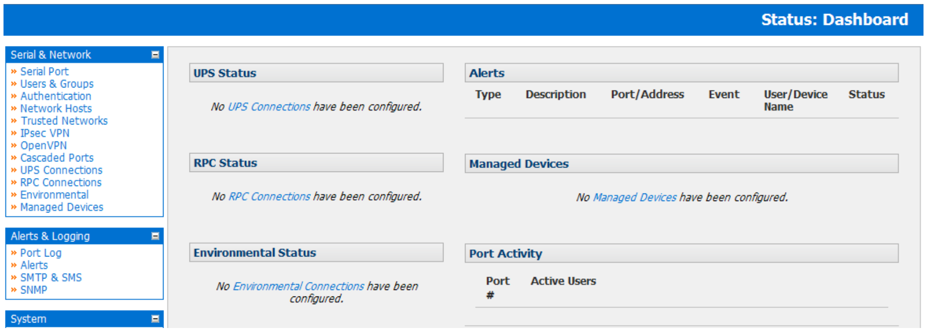
```
<27>Dec 22 20:36:32 stunnel: LOG3[1482:0]: SSL_accept: Peer suddenly
disconnected
<27>Dec 22 20:36:32 stunnel: LOG3[1485:0]: SSL_accept: Peer suddenly
disconnected
```

To make it easier to find information in the local Syslog file, a pattern matching filter tool is provided.

- Specify the **Match Pattern** that is to be searched for (e.g. the search for *mount* is shown below) and click **Apply**. The Syslog will then be represented with only those entries that actually include the specified pattern

12.5 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 groups.



UPS Status

No UPS Connections have been configured.

Alerts

Type	Description	Port/Address	Event	User/Device Name	Status

RPC Status

No RPC Connections have been configured.

Managed Devices

No Managed Devices have been configured.

Environmental Status

No Environmental Connections have been configured.

Port Activity

Port #	Active Users

12.5.1 Configuring the Dashboard

Only users who are members of the admin group (and the root user) can configure and access the dashboard. To configure a custom dashboard:

- Select **System: Configure Dashboard** and select the user (or group) you are configuring this custom dashboard layout for

Note: *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.*

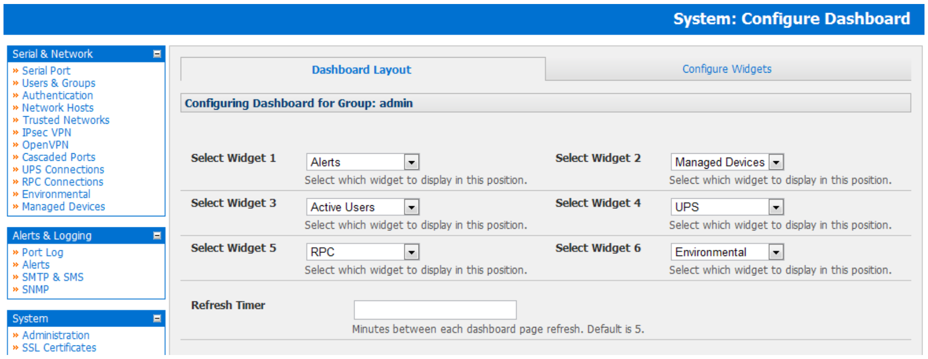
*If there is no dashboard layout configured for John but there is an **admin** group dashboard configured then you will see the admin group dashboard instead. If there is no user dashboard or admin group dashboard configured, then you will see the default dashboard.*

*The **root** user does not have its own dashboard.*

The above configuration options are intended to enable admin users to setup their own custom dashboards

The Dashboard displays *six widgets*. These widgets include each of the Status screens (alerts, devices, ports ups, rpc and environmental status) and a custom script screen. The *admin* user can configure which of these widgets is to be displayed where:

- Go to the **Dashboard Layout** panel and select which widget is to be displayed in each of the six display locations (widget1 …6)
- Click Apply



System: Configure Dashboard

Configuring Dashboard for Group: admin

Select Widget 1: Alerts
Select which widget to display in this position.

Select Widget 2: Managed Devices
Select which widget to display in this position.

Select Widget 3: Active Users
Select which widget to display in this position.

Select Widget 4: UPS
Select which widget to display in this position.

Select Widget 5: RPC
Select which widget to display in this position.

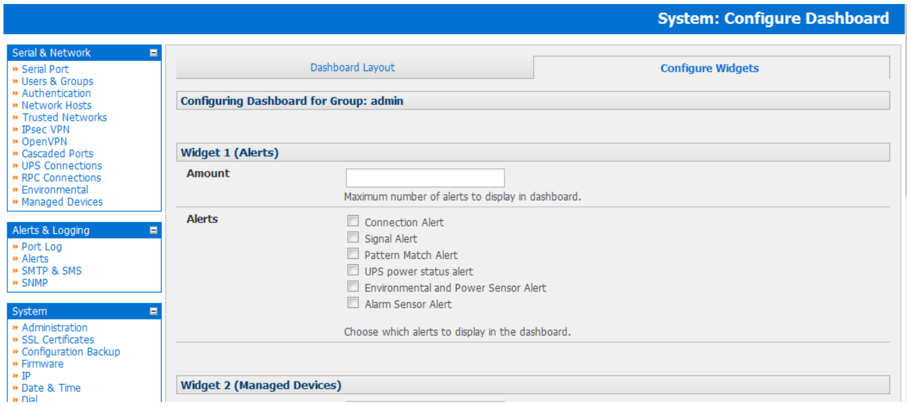
Select Widget 6: Environmental
Select which widget to display in this position.

Refresh Timer:
Minutes between each dashboard page refresh. Default is 5.

Note: *The Alerts widget is a new screen that shows the current alerts status. When an alert gets triggered, a corresponding .XML file is created in **/var/run/alerts/**. The dashboard scans all these files and displays a summary status in the alerts widget. When an alert is deleted the corresponding .XML files that belong to that alert are also deleted.*

To configure what is to be displayed by each widget:

- Go to the **Configure Widgets** panel and configure each selected widget (e.g. specify which UPS status is to be displayed on the *ups widget* or the maximum number of Managed Devices to be displayed in the *devices widget*
- Click Apply



System: Configure Dashboard

Configuring Dashboard for Group: admin

Widget 1 (Alerts)

Amount

Maximum number of alerts to display in dashboard.

Alerts

☐ Connection Alert
☐ Signal Alert
☐ Pattern Match Alert
☐ UPS power status alert
☐ Environmental and Power Sensor Alert
☐ Alarm Sensor Alert

Choose which alerts to display in the dashboard.

Widget 2 (Managed Devices)

***Note**: Dashboard configuration is stored in the /etc/config/config.xml file. Each configured dashboard will increase the config file. If this file gets too big, you can run out of memory space on the Console Server.*

12.5.2 Creating custom widgets for the Dashboard

To run a custom script inside a dashboard widget:

Create a file called "*widget-<name>.sh*" in the folder */etc/config/scripts/* where *<name>* can be anything. You can have as many custom dashboard files as you want.

Inside this file you can put any code you wish. When configuring the dashboard, choose "*widget-<name>.sh*" in the dropdown list. The dashboard will run the script and display the output of the script commands directly on the screen, inside the specific widget.

The best way to format the output would be to send HTML commands back to the browser by adding echo commands in the script:

```
echo '<table>'
```

You can of course run any command and its output will be displayed in the widget window directly.
Below is an example script which 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 specific 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://www.vinras.com/images/linux-online-inc.jpg">'
echo '</td></tr>'
echo '</table>'

exit 0
```

Chapter 13: Management

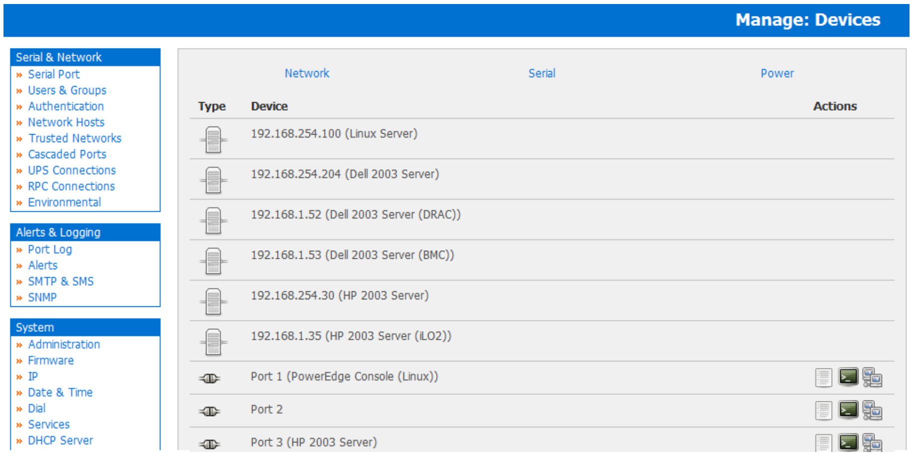
The Console Server has a small number of **Manage** reports and tools that are available to both Administrators and Users:

- Access and control authorized devices
- View serial port logs and host logs for those devices
- Use SDT Connector or the Web Terminal to access serially attached consoles
- Power control

13.1 Device Management

To display all the connected Serial devices, Network Hosts and Power devices:

- Select **Manage: Devices**. By selecting the **Serial/ Network/ Power** item, the display will be reduced to only those devices



Type	Device	Actions
	192.168.254.100 (Linux Server)	
	192.168.254.204 (Dell 2003 Server)	
	192.168.1.52 (Dell 2003 Server (DRAC))	
	192.168.1.53 (Dell 2003 Server (BMC))	
	192.168.254.30 (HP 2003 Server)	
	192.168.1.35 (HP 2003 Server (iLO2))	
	Port 1 (PowerEdge Console (Linux))	
	Port 2	
	Port 3 (HP 2003 Server)	

The user can take a range of actions on each of these Serial/Network/Power devices by selecting the **Action** icon or the related Manage menu item. Selecting the Manager Power icon or the **Manage: Power** menu is covered in *Chapter 8*.

13.2 Port and Host Log Management

Administrators and Users can view logs of data transfers to connected devices.

- Select **Manage: Port Logs** and the serial Port # to be displayed
- To display Host logs select **Manage: Host Logs** and the Host to be displayed

13.3 Terminal Connection

There are two methods available for accessing the console server command line and devices attached to the console server serial ports, directly from a web browser:

- The Web Terminal service uses AJAX to enable the web browser to connect to the console server using HTTP or HTTPS, as a terminal - without the need for additional client installation on the user's PC
- The SDT Connector service launches a pre-installed SDT Connector client on the user's PC to establish secure SSH access, then uses pre-installed client software on the client PC to connect to the console server

Web browser access is available to users who are a member of the admin or users groups.

13.3.1 Web Terminal

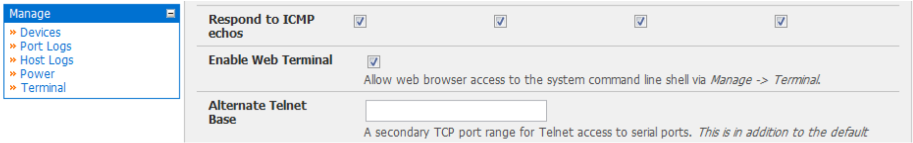
The AJAX based Web Terminal service may be used to access the console server command line or attached serial devices.

***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 Management Console, i.e. if you are browsing using an https:// URL (this is the default), the Web Terminal connects using HTTPS.*

13.3.1.1 Web Terminal to Command Line

To enable the Web Terminal service for the console server

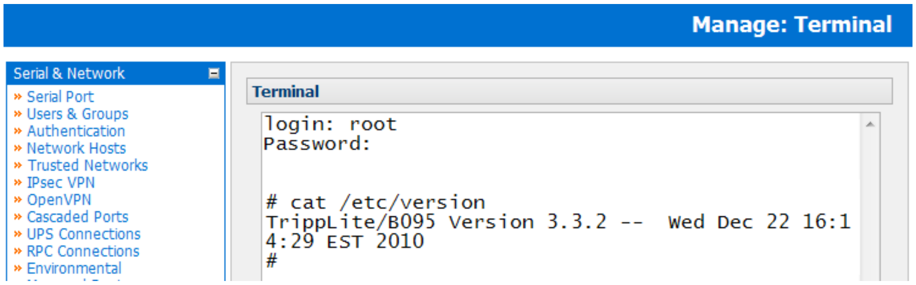
- Select the tab in the **System: Firewall** menu
- Check **Enable Web Terminal** and click **Apply**



Respond to ICMP echos	☑	☑	☑	☑
Enable Web Terminal	☑ Allow web browser access to the system command line shell via *Manage -> Terminal.*			
Alternate Telnet Base	A secondary TCP port range for Telnet access to serial ports. *This is in addition to the default*			

Administrators can now communicate directly with the Console Server command line from their browser:

- Select **Manage: Terminal** to display the Web Terminal from which you can log in to the Console Server command line

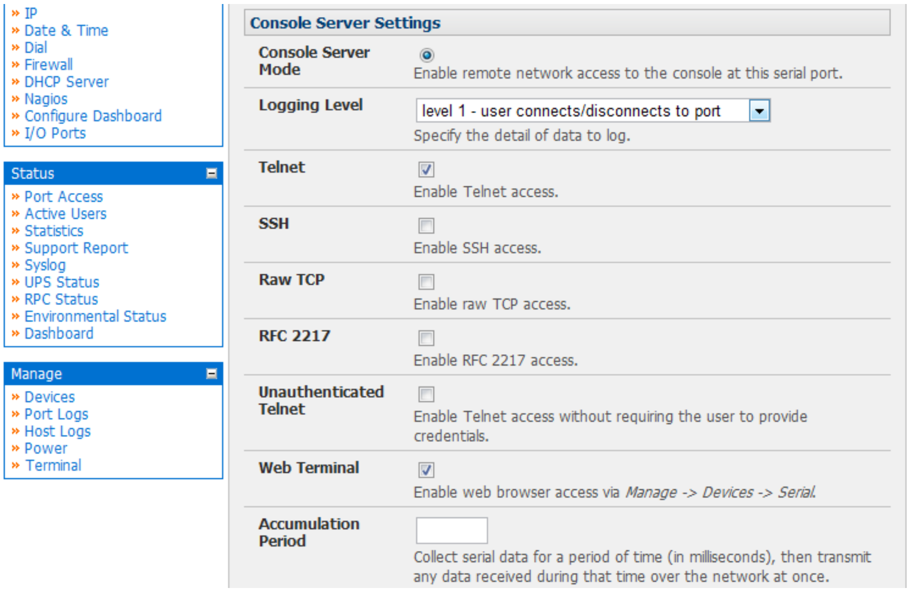


```
login: root
Password:


# cat /etc/version
TrippLite/B095 Version 3.3.2 --  Wed Dec 22 16:1
4:29 EST 2010
#
```

13.3.1.2 Web Terminal to Serial Device

To enable the Web Terminal service for each serial port you want to access:

- Select **Serial & Network: Serial Port** and click **Edit**. Ensure the serial port is in Console Server *Mode*
- Check **Web Terminal** and click **Apply**

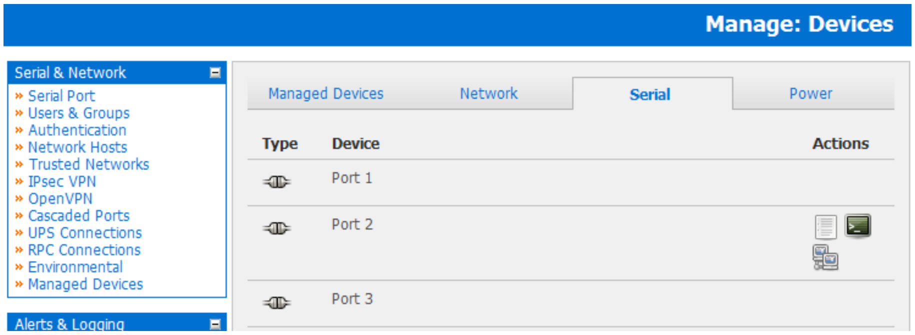


Console Server Settings

Console Server Mode	● Enable remote network access to the console at this serial port.
Logging Level	level 1 - user connects/disconnects to port ▾ Specify the detail of data to log.
Telnet	☑ Enable Telnet access.
SSH	☐ Enable SSH access.
Raw TCP	☐ Enable raw TCP access.
RFC 2217	☐ Enable RFC 2217 access.
Unauthenticated Telnet	☐ Enable Telnet access without requiring the user to provide credentials.
Web Terminal	☑ Enable web browser access via *Manage -> Devices -> Serial.*
Accumulation Period	Collect serial data for a period of time (in milliseconds), then transmit any data received during that time over the network at once.

Administrator and Users can communicate directly with serial port attached devices from their browser:

- Select the **Serial** tab on the **Manage: Devices** menu
- Under the *Action* column, click the **Web Terminal** icon to display the Web Terminal, connected directly to the attached serial device



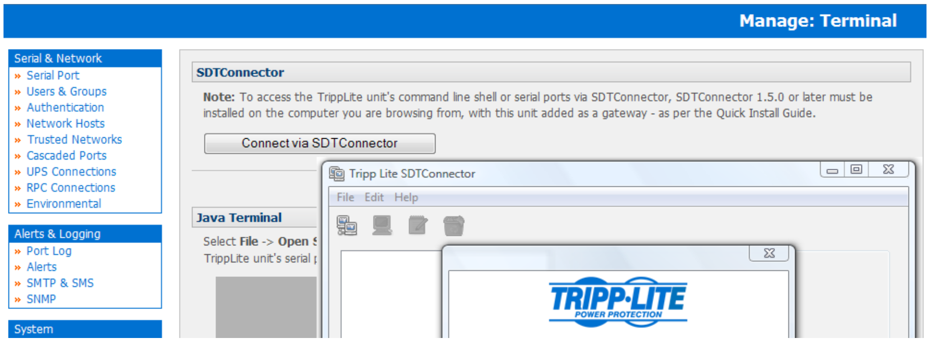
Type	Device	Actions
	Port 1	
	Port 2	
	Port 3	

***Note**: The Web Terminal feature was introduced in firmware V3.3.2. Earlier releases had an open source **jcterm** java terminal applet which could be downloaded into your browser to connect to the Console Server and attached serial port devices. However **jcterm** had some JRE compatibility issues and is no longer supported*

13.3.2 SDTConnector access

Administrator 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 tenet client, or using a Web terminal and their browser

- Select **Manage: Terminal**
- Click the **Connect to SDT Connector** button. This will to activate the SDT Connector client on the computer you are browsing and load your local telnet client to connect to the command line or serial port using SSH



Note: To access the TrippLite unit's command line shell or serial ports via SDTConnector, SDTConnector 1.5.0 or later must be installed on the computer you are browsing from, with this unit added as a gateway - as per the Quick Install Guide.

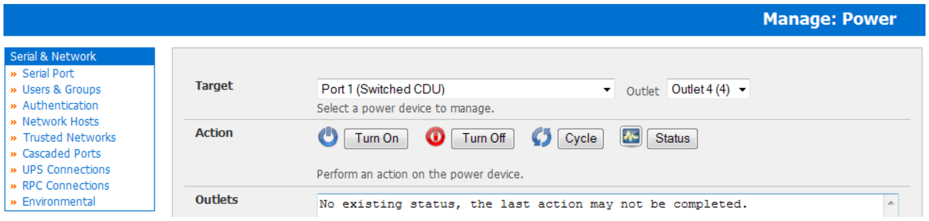
Select **File** -> **Open S**
TrippLite unit's serial p

***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 6*

13.4 Power Management

Administrators and Users can access and manage the connected power devices.

- Select **Manage: Power**



Target Port 1 (Switched CDU) Outlet Outlet 4 (4)
Select a power device to manage.

Action
Perform an action on the power device.

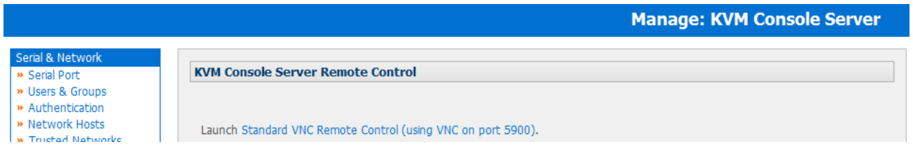
Outlets

```
No existing status, the last action may not be completed.
```

13.5 Remote Console Access (B092-016 only)

Administrator and Users can also connect to the B092-016 Console Server with PowerAlert remotely (as if they were plugged in locally to the KVM connectors on the B092-016). This connection will enable the remote users to run the PowerAlert software and the other thin client programs (refer to Chapter 16) embedded in the Console Server:

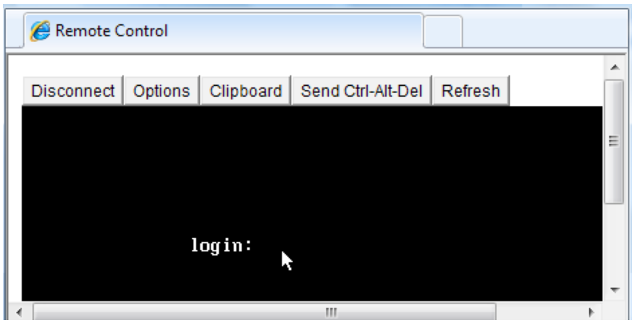
- Select **Manage: KVM Console Server**



KVM Console Server Remote Control

Launch Standard VNC Remote Control (using VNC on port 5900).

- Click **Standard VNC Remote control** and a VNC Java applet will be loaded into your browser to connect to the B092-016 Console Server. Then log in to the VNC applet and the Console Server (refer to Chapter 16.3 for more details)



```
login:
```

Chapter 14: Command Line Configuration

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 and configure the ports etc.

This *config* documentation in this chapter walks thru 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 refer to the next chapter

When displaying a command, the convention used in the rest of this chapter is to use single quotes (") for user defined values (e.g. descriptions and names). Element values without single quotes must 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.

14.1 Accessing *config* from the command line

The Console Server runs a standard Linux kernel and embeds a suite of open source applications. So 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*. However without care these configurations may not withstand a *power-cycle-reset* or *reconfigure*.

So Tripp Lite 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 local DB-9 console port and terminal device. Configure the serial connection of the terminal device you are using to 115200bps, 8 data bits, no parity and one stop bit

  - If you are connecting over the LAN then you will need to 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*. You should now see the command line prompt which is a hash (#)

This chapter is not intended to teach you Linux. We assume you already have a certain level of understanding before you execute Linux kernel level commands.

The *config* tool

Syntax

config [-ahv] [-d id] [-g id] [-p path] [-r configurator] [-s id=value] [-P id]

Description

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* which 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 user '*root*' can configure from the shell.

By default, the *config* elements are separated by a '.' character. The root of the *config* tree is called <*config*>. To address a specific element place a '.' between each node/branch e.g. 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 path */bin/config*.

Options

-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, 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 15. You can also assign the config element to "", or delete the entire config node using *-d*:

```
# /bin/config -d 'element name'
```

All passwords are saved in plaintext except the user passwords and the system passwords, which are encrypted.

Note*: 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 refer Chapter 15 Advanced Configuration.

14.1.1 Serial Port configuration

The first set of configurations that needs to be made to any serial port are the RS232 common settings. For example to setup 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
```

Note: Supported serial port baud-rates are ‘50’, ‘75’, ‘110’, ‘134’, ‘150’, ‘200’, ‘300’, ‘600’, ‘1200’, ‘1800’, ‘2400’, ‘4800’, ‘9600’, '19200', '38400', '57600', '115200', and '230400'.
Supported parity values are 'None', 'Odd', 'Even', 'Mark' and 'Space'.
Supported data-bits values are '8', '7', '6' and '5'.
Supported stop-bits values are '1', '1.5' and '2'.
Supported flow-control values are 'Hardware', 'Software' and '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 (refer Chapter 4 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	*100 ms*
Escape character	*% (default is ~)*
log level	*2 (default is 0)*
Shell power command menu	*Enabled*
RFC2217 access	*Enabled*
Limit pot to 1 connection	*Enabled*
SSH access	*Enabled*
TCP access	*Enabled*
telnet access	*Disabled*
Unauthorized telnet access	*Disabled*

```
# 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'
'facility' can be:
        Default
        local 0-7
        auth
        authpriv
        cron
        daemon
        ftp
        kern
        lpr
        mail
        news
        user
        uucp
# config -s config.ports.port#.syslog.priority='priority'
'priority' can be:
        Default
        warning
        notice
        Info
        error
        emergency
        debug
        critical
        alert
```

14.1.2 Adding and removing Users

Firstly, determine the total number of existing Users (if you have no existing Users you can assume this is 0):

```
# config -g config.users.total
```

This command should display *config.users.total 1*. Note that 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.

To add a user (with Username=John, Password=secret and Description =mySecondUser) issue the commands:

```
# config -s config.users.total=2 (assuming we already have 1 user configured)
# 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. In fact, 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 re-set.

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='' (the 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 (total number of users that have access to this outlet)
```

If more users are given access to this power outlet, then increment the 'config.ports.port1.power.outlet3.users.total' element 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 (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 (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 i.e. 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:

```
# ./delete-node config.users.user2
```

The following command will synchronize the live system with the new configuration:

```
# config -r users
```

14.1.3 Adding and removing User Groups

The Console Server is configured with a few default user groups (even though 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 you'd 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 manager, 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 (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 (total number of groups having 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 (total number of users having access to host)
```

To delete the group called Group7, use the following command:

```
# rmuser Group7
```

Attention: The *rmuser* script is a generic script to remove any config element from config.xml correctly. However, 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
```

14.1.4 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' (list of remote authentiction and authorization servers.)

# config -s config.auth.tacacs.acct_server='comma separated list' (list of remote accounting servers. If unset, Authentication and Authorization Server Address will be used.)

# config -s config.auth.tacacs.password='password'

To configure RADIUS authentication:

# config -s config.auth.radius.auth_server='comma separated list' (list of remote authentiction and authorization servers.)

# config -s config.auth.radius.acct_server='comma separated list' (list of remote accounting servers. If unset, Authentication and Authorization Server Address will be used.)

# config -s config.auth.radius.password='password'

To configure LDAP authentication:

# config -s config.auth.ldap.server='comma separated list' (list of remote servers.)

# config -s config.auth.ldap.basedn='name' (The distinguished name of the search base. For example: dc=my-company,dc=com)

# config -s config.auth.ldap.binddn='name' (The distinguished name to bind to the server with. The default is to bind anonymously.)

# config -s config.auth.radius.password='password'

The following command will synchronize the live system with the new configuration:

# config -r auth

14.1.5 Network Hosts

To determine the total number of currently configured hosts:

```
# config -g config.sdt.hosts.total
```

Assume this value is equal to 3. If you add another host, make sure to 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* then assume 0 hosts are configured.

Add power device host

To add a UPS/RPC network host with the following details:

IP address/ 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 commands below:

```
# 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
```

The *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:

IP address/ DNS name	192.168.3.10
Host name	OfficePC
Description	MyPC
Allowed sevices	ssh port 22,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='' (leave this value blank)
# 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
```

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 -hosts
```

14.1.6 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*

Note that if you see *config.portaccess.total* this means you have 0 rules configured.

Your new rule will be the existing total plus 1. So if the previous command gave you 0 then you start with rule number 1. If you already have 1 rule your new rule will be number 2 etc.

If you want to restrict access to serial port 5 to computers from a single class C network (*192.168.5.0* say) you need to issue the following commands (assuming you have a previous rule in place).

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
```

14.1.7 Cascaded Ports

To add a new slave device with the following settings:

IP address/DNS name	192.168.0.153
Description	CM in office 42
Label	BL6-5
Number of ports	16

The following commands must be issued:

```
# config -s config.cascade.slaves.slave1.address=192.168.0.153
# config -s "config.cascade.slaves.slave1.description=CM in office 42"
# config -s config.cascade.slaves.slave1.label=BL6-5
# 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, then the master SSH public key will need to be manually copied to every slave device before cascaded ports will work (refer *Chapter 4*)

The following command will synchronize the live system with the new configuration:

```
# config -r cascade
```

14.1.8 UPS Connections

Managed UPS Systems

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 to 'ups'.

To add a managed UPS with the following values:

Connected via	Port 1
UPS name	My UPS
Description	UPS in room 5
Username to connect to UPS	User2
Password to connect to UPS	secret
shutdown order	2 (0 shuts down first)
Driver	genericups
Driver option - option	option
Driver option - argument	argument
Logging	Enabled
Log interval	2 minutes
Run script when power is critical	Enabled

```
# config -s config.ups.monitors.monitor1.port=/dev/port01
```

If the port number is higher than 9, eg port 13, enter:

```
# config -s config.ups.monitors.monitor1.port=/dev/port13

# config -s "config.ups.monitors.monitor1.name=My UPS"
# config -s "config.ups.monitors.monitor1.description=UPS in room 5"
# config -s config.ups.monitors.monitor1.username=User2
# config -s config.ups.monitors.monitor1.password=secret
# 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
```

Make sure to increment the total monitors:

```
# config -s config.ups.monitors.total=1
```

The 5 commands below will add the UPS to 'Managed devices. Assuming there are already 2 managed devices configured:

```
# 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=UPS in toom 5"
# config -s config.devices.total=3
```

To delete this managed UPS:

```
# config -d config.ups.monitors.monitor1
```

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):

UPS name	oldUPS
Description	UPS in room 2
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=UPS in room 2"
# 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
```

14.1.9 RPC Connections

You can add an RPC connection from the command line but it is not recommended that you do so because of dependency issues.

However FYI 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'.

To add an RPC with the following values:

RPC type	APC 7900
Connected via	Port 2
UPS name	MyRPC
Description	RPC in room 5
Login name for device	rpclogin
Login password for device	secret
SNMP community	v1 or v2c
Logging	Enabled
Log interval	600 second
Number of power outlets	4 (depends on the type/model of the RPC)

```
# config -s config.ports.port2.power.type=APC 7900
# config -s config.ports.port2.power.name=MyRPC
# config -s "config.ports.port2.power.description=RPC in room 5"
# config -s config.ports.port2.power.username=rpclogin
# config -s config.ports.port2.power.password=secret
# 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=RPC in room 5"
# config -s config.devices.total=3
```

The following command will synchronize the live system with the new configuration:

```
# config -a
```

14.1.10 Environmental

To configure an environmental monitor with the following details:

Monitor name	Envi4
Monitor Description	Monitor in room 5
Temperature offset	2
Humidity offset	5
Enable alarm 1 ?	yes
Alarm 1 label	door alarm
Enable alarm 2 ?	yes
Alarm 2 label	window alarm
Logging enabled ?	yes
Log interval	120 seconds

```
# config -s config.ports.port3.enviro.name=Envi4
# config -s "config.ports.port3.enviro.description=Monitor in room 5"
# config -s config.ports.port3.enviro.offsets.temp=2
# config -s config.ports.port3.enviro.offsets.humid=5
# config -s config.ports.port3.enviro.alarms.alarm1.alarmstate=on
# config -s config.ports.port3.enviro.alarms.alarm1.label=door alarm
# config -s config.ports.port3.enviro.alarms.alarm2.alarmstate=on
# config -s config.ports.port3.enviro.alarms.alarm2.label=window alarm
# config -s config.ports.port3.enviro.alarms.total=2
# config -s config.ports.port3.enviro.log.enabled=on
# config -s config.ports.port3.enviro.log.interval=120
```

It is important to assign alarms.total=2 even if they are off.

The following 5 commands will add the environmental monitor to 'Managed devices':

To get the total number of managed devices:

```
# config -g config.devices.total
```

Make sure to use the total + 1 for the new device below:

```
# config -s config. devices.device5.connections.connection1.name=Envi4
# config -s "config. devices.device5.connections.connection1.type=EMD Unit"
# config -s config. devices.device5.name=Envi4
# config -s "config. devices.device5.description=Monitor in room 5"
# config -s config.devices.total=5
```

The following command will synchronize the live system with the new configuration:

```
# config -a
```

14.1.11 Managed Devices

To add a managed device: (also see UPS, RPC connections and Environmental)

```
# config -s "config.devices.device8.name=my device"
# config -s "config.devices.device8.description=The eighth device"
# config -s "config.devices.device8.connections.connection1.name=my device"
# config -s config.devices.device8.connections.connection1.type=[serial | Host | UPS | RPC]
# config -s config.devices.total=8          (decrement this value when deleting a managed device)
```

To delete the above managed device:

```
# config -d config.devices.device8
```

The following command will synchronize the live system with the new configuration:

```
# config -a
```

14.1.12 Port log

To configure serial/network port logging:

```
# config -s config.eventlog.server.address='remote server ip address'
# config -s config.eventlog.server.logfacility='facility'
```

'facility' can be:

- *Daemon*
- *Local 0-7*
- *Authentication*
- *Kernel*
- *User*
- *Syslog*
- *Mail*
- *News*
- *UUCP*

```
# config -s config.eventlog.server.logpriority='priority'
```

'*priority*' can be:

- *Info*
- *Alert*
- *Critical*
- *Debug*
- *Emergency*
- *Error*
- *Notice*
- *Warning*

Assume the remote log server needs a username 'name1' and password 'secret':

```
# config -s config.eventlog.server.username=name1
# config -s config.eventlog.server.password=secret
```

To set the remote path as '/tripplite/logs' to save logged data:

```
# config -s config.eventlog.server.path=/tripplite/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
```

14.1.13 Alerts

You can add an email, SNMP or NAGIOS alert by following the steps below.

The general settings for all alerts

Assume this is our second alert, and we want to send alert emails to john@company.com and sms's to peter@compnany.com:

```
# config -s config.alerts.alert2.description=MySecondAlert
# config -s config.alerts.alert2.email=john@company.com
# config -s config.alerts.alert2.email2=peter@company.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
```

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='host name'
# 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
```

Environmental and Power Sensor Alert

```
# config -s config.alerts.alert2.enviro.high.critical='critical value'
# config -s config.alerts.alert2.enviro.high.warning='warning value'
# config -s config.alerts.alert2.enviro.hysteresis='value'
# config -s config.alerts.alert2.enviro.low.critical='critical value'
# config -s config.alerts.alert2.enviro.low.warning='warning value'
# config -s config.alerts.alert2.enviro1='Enviro sensor name'
# config -s config.alerts.alert2.outlet#='RPCname'.outlet#
```

'alert2.outlet#' increments sequentially with each added outlet. The second 'outlet#' refers to the specific RPC power outlets.

```
# config -s config.alerts.alert2.rpc#='RPC name'
# config -s config.alerts.alert2.sensor=[ temp | humid | load | charge]
# config -s config.alerts.alert2.signal=DSR
# config -s config.alerts.alert2.type=enviro
# config -s config.alerts.alert2.ups1='UPSname@hostname'
```

Example1: To configure a temperature sensor alert for a sensor called 'SensorInRoom42':

```
# config -s config.alerts.alert2.sensor=temp
# config -s config.alerts.alert2.enviro.high.critical=60
# config -s config.alerts.alert2.enviro.high.warning=50
# config -s config.alerts.alert2.enviro.hysteresis=2
# config -s config.alerts.alert2.enviro.low.critical=5
# config -s config.alerts.alert2.enviro.low.warning=10
# config -s config.alerts.alert2.enviro1=SensorInRoom42
# config -s config.alerts.alert2.signal=DSR
# config -s config.alerts.alert2.type=enviro
```

Example2: To configure a load sensor alert for outlets 2 and 4 for an RPC called 'RPCInRoom20':

```
# config -s config.alerts.alert2.outlet1='RPCname'.outlet2
# config -s config.alerts.alert2.outlet2='RPCname'.outlet4
# config -s config.alerts.alert2.enviro.high.critical=300
# config -s config.alerts.alert2.enviro.high.warning=280
# config -s config.alerts.alert2.enviro.hysteresis=20
# config -s config.alerts.alert2.enviro.low.critical=50
# config -s config.alerts.alert2.enviro.low.warning=70
# config -s config.alerts.alert2.rpc1=RPCInRoom20
# config -s config.alerts.alert2.sensor=load
# config -s config.alerts.alert2.signal=DSR
# config -s config.alerts.alert2.type=enviro
```

Alarm Sensor Alert

To set an alert for 'doorAlarm' and 'windowAlarm' which are two alarms connected to an environmental sensor called 'SensorInRoom3'. Both alarms are disabled on Mondays from 8:15am to 2:30pm:

```
# 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
```

14.1.14 SMTP & SMS

To set-up an SMTP mail or SMS server with the following details:

Outgoing server address	mail.company.com
Secure connection type	SSL
Sender	John@company.com
Server username	john
Server password	secret
Subject line	SMTP alerts

```
# config -s config.system.smtp.server=mail.company.com
# config -s config.system.smtp.encryption=SSL  (can also be TLS or None )
# config -s config.system.smtp.sender=John@companycom
# config -s config.system.smtp.username=john
# config -s config.system.smtp.password=secret
# 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.company.com
# config -s config.system.smtp.encryption2=SSL  (can also be TLS or None )
# config -s config.system.smtp.sender2=John@company.com
# config -s config.system.smtp.username2=john
# config -s config.system.smtp.password2=secret
# config -s config.system.smtp.subject2=SMTP alerts
```

The following command will synchronize the live system with the new configuration:

```
# config -a
```

14.1.15 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'   (default is 162)
# config -s config.system.snmp.address='NMS IP network address'
# config -s config.system.snmp.commnity='community name'  (v1 and v2c only)
# config -s config.system.snmp.engineid='ID'   (v3 only)
# config -s config.system.snmp.username='username'   (v3 only)
# config -s config.system.snmp.password='password'  (v3 only)
# config -s config.system.snmp.version=[ 1 | 2c | 3 ]
```

The following command will synchronize the live system with the new configuration:

```
# config -a
```

14.1.16 Administration

To change the administration settings to:

System Name	og.mydomain.com
System Password (root account)	secret
Description	Device in office 2

```
# config -s config.system.name=og.mydomain.com
# config -P config.system.password              (will prompt user for a password)
# config -s "config.system.location=Device in office 2"
```

NOTE: The -P parameter will prompt the user for a password, and encrypt it. In fact, 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 re-set.

The following command will synchronize the live system with the new configuration:

```
# config –a
```

14.1.17 IP settings

To configure the primary network interface with static settings:

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=[ Auto | 100baseTx-FD | 100baseTx-HD | 10baseT-HD ] 10baseT-FD
```

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 configure the management lan interface, use the same commands as above but replace:

config.interfaces.wan, with config.interfaces.lan

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=[ eth1 | console | modem ]
```

The network interfaces can also be configured automatically:

```
# config -s config.interfaces.wan.mode=dhcp
# config -s config.interfaces.lan.mode=dhcp
```

The following command will synchronize the live system with the new configuration:

```
# /bin/config –-run=ipconfig
```

The following command will synchronize the live system with the new configuration:

```
# config -r ipconfig
```

14.1.18 Date & 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 092216452005.05              Format is MMDDhhmm[[CC]YY][.ss]
```

Then the following command will save this new system time to the hardware clock:

```
# /bin/hwclock -systohc
```

Alternatively, to change the hardware clock:

```
# /bin/hwclock -- set --date=092216452005.05        Format is MMDDhhmm[[CC]YY][.ss]
```

Then the following command will save this new hardware clock time as the system time:

```
# /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
```

14.1.19 Dial-in settings

To enable dial-in access on the DB9 serial port from the command line with the following attributes:

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	0800223665
User to dial as	user1
Password for user	secret

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=secret
```

To make the dialed connection the default route:

```
# config -s config.console.ppp.defaultroute=on
```

Please note that supported authentication types are 'None', 'PAP', 'CHAP' and 'MSCHAPv2'.
Supported serial port baud-rates are '9600', '19200', '38400', '57600', '115200', and '230400'.
Supported parity values are 'None', 'Odd', 'Even', 'Mark' and 'Space'.
Supported data-bits values are '8', '7', '6' and '5'.
Supported stop-bits values are '1', '1.5' and '2'.
Supported flow-control values are 'Hardware', 'Software' and 'None'.

If you do not wish to use out-of-band dial-in access please note that the procedure for enabling start-up messages on the console port is covered in Chapter 15 - Accessing the Console Port.

The following command will synchronize the live system with the new configuration:

```
# config –a
```

14.1.20 DHCP server

To enable the DHCP server on the console management LAN, with settings:

Default lease time	200000 seconds
Maximum lease time	300000 seconds
DNS server1	192.168.2.3
DNS server2	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

Issue the 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 –a
```

14.1.21 Services

You can manually enable or disable network servers from the command line. For example if you wanted to guarantee the following server configuration:

HTTP Server	Enabled
HTTPS Server	Disabled
Telnet Server	Disabled
SSH Server	Enabled
SNMP Server	Disabled
Ping Replies (Respond to ICMP echo requests)	Disabled
TFTP server	Enabled

```
# 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
```

To set secondary port ranges for any service

```
# config -s config.services.telnet.portbase='port base number'          Default: 2000
# config -s config.services.ssh.portbase='port base number'             Default: 3000
# config -s config.services.tcp.portbase='port base number'             Default: 4000
# config -s config.services.rfc2217.portbase='port base number'         Default: 5000
# config -s config.services.unauthtel.portbase='port base number        Default: 6000
```

The following command will synchronize the live system with the new configuration:

```
# config -a
```

14.1.22 NAGIOS

To configure NAGIOS with the following settings:

NAGIOS host name	b095 (Name of this system)
NAGIOS host address	192.168.0.1 (IP to find this device at)
NAGIOS server address	192.168.0.10 (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)

```
# config -s config.system.nagios.enabled=on
# config -s config.system.nagios.name=b095
# 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       (disables SDT for nagios extensions)
# config -s config.system.nagios.sdt.address=192.168.0.1
# config -s config.system.nagios.nrpe.prefer=''
```

To configure NRPE with following settings:

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

```
# 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:

NSCA encryption	BLOWFISH (can be: [None | XOR | DES | TRPLEDES | CAST-256 | BLOWFISH | TWOFISH | RIJNDAEL-256 | SERPENT | GOST]
NSCA password	secret
NSCA check-in interval	5 minutes
NSCA port	5650 (defaults to 5667)
user to run as	User1 (defaults to nsca)
group to run as	Group1 (defaults to nobody)

```
# 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 –a
```

14.2 General Linux command usage

The Console Server platform is a dedicated Linux computer, optimized to provide access to serial consoles of critical server systems and control network connected hosts. Being based around uClinux (a small footprint but extensible Linux), it embodies a myriad of popular and proven Linux software modules for networking (NetFilter, IPTables), secure access (OpenSSH) and communications (OpenSSL) and sophisticated user authentication (PAM, RADIUS, TACACS+ and LDAP).

Many components of the Console Server software are licensed under the GNU General Public License (version 2). You may obtain a copy of the GNU General Public License at http://www.fsf.org/copyleft/gpl.html and source code will provided for any of the components of the Software licensed under the GNU General Public License upon request. The Console Servers are built on the 2.6 uClinux kernel as developed by the uClinux project. This is GPL code and source can be found: http://cvs.uclinux.org.

Supported commands that have config files that can be altered include:

portmanager
inetd
init
ssh/sshd/scp/sshkeygen
ucd-snmpd
samba
fnord (web server)
sslwrap

Commands you can run from the command line on the Console Server include::

loopback
bash (shell)
busybox http://www.busybox.net/downloads/BusyBox.html (has lots of unix shell commands and tools)
chat
dhcpcd
ftp
hd
hwclock
iproute
iptables
netcat
ifconfig
mii-tool
netstat
route
openntpd
ping
portmap
pppd
routed
setserial
smtpclient
stty
stunel
tcpdump
tftp
tip
traceroute

More details on the above Linux commands can found online at:

http://en.tldp.org/HOWTO/HOWTO-INDEX/howtos.html
http://www.faqs.org/docs/Linux-HOWTO/Remote-Serial-Console-HOWTO.html
http://www.stokely.com/unix.serial.port.resources/serial.switch.html

Chapter 15: Advanced Configuration

Console Servers run the embedded Linux operating system. So 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 14*).

The Linux kernel in the Console Server also supports GNU bash shell script 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 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*
- sms server tools
- disable multicasting

15.1 Custom Scripting

The Console Server supports GNU bash shell commands (refer Appendix A) enabling the Administrator to run custom scripts.

15.1.1 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 e.g. if you wanted to display *hello world:*

```
#!/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.

15.1.2 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) or */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 a 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 of 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. 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   (if the directory does not already exist)
# cp /etc/scripts/portmanager-pattern-alert /etc/config/scripts/portmanager-pattern-alert
```

The next step will be to edit the new script file. Firstly, open the file */etc/config/scripts/portmanager-pattern-alert* using vi (or any other editor), and 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.

15.1.3 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 and which 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, in order 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.

15.1.4 Example script - Multiple email 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* which gets run from within all the alert scripts (e.g. *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. In particular let's 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.

15.1.5 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 (e.g. "*config.users.user1*" or "*config.sdt.hosts.host1*").

So *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 user 3:

```
    # ./delete-node config.users.user3
```

The *delete-node* script

```
    #!/bin/bash
    #User must provide the node to be removed. e.g. "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 is the last field in the node path e.g. "user1"
# ROOTNODE is the upper level of the node e.g. "config.users"
# NUMBER is the integer value extracted from LASTFIELD e.g. "1"
# TOTALNODE is the node name for the total e.g. "config.users.total"
# TOTAL is the value of the total number of items before deleting e.g. "3"
# NEWTOTAL is the 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
```

15.1.6 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/ 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. The *ping-detect* is run from */etc/config/rc.local* to make sure that the monitoring starts whenever the system boots.

So if 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
- 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 "&".

Remember 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 *ping-detect* script

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:

```
#!/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
```

15.1.7 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 e.g. 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 as if another user/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.

15.1.8 Backing-up the configuration and restoring using a local USB stick

The ***/etc/scripts/backup-usb*** script been written to save and load custom configuration using a USB flash disk. Before saving 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]

COMMAND:

```
check-magic -- check volume label
set-magic -- set volume label
save [FILE] -- save configuration to USB
delete [FILE] -- delete a configuration tarbal 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 at all.

15.1.9 Backing-up the configuration off-box

If you do not have a USB 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, as 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/b095.config # umount/mnt/
```

Example transfer off-box via scp:

```
# config -e /tmp/b095.config
# scp /tmp/b095.config 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.

As SSH uses these keys to avoid man-in-the-middle attacks, logging in may be disrupted.

15.2 Advanced Portmanager

The *portmanger* program manages the Console Server serial ports. It routes network connection to serial ports, checks permissions, and monitors and logs all the data flowing to/from the ports.

15.2.1 Portmanager commands

pmshell

The *pmshell* command acts similar to the standard *tip* or *cu* commands, but all serial port access is directed *via* the portmanager.

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/cu support. For SSH you must prefix the escape with an additional '~' command (i.e. use the '~~' escape)

Send Break: Typing the character sequence '~b' will generate a BREAK on the serial port (if you're doing this over ssh, you'll need to type "~~b")

History: Typing the character sequence '~h' will generate a history on the serial port.

Quit pmshell: Typing the character sequence '~.' will exit from pmshell.

Set RTS to 1 run the command: pmshell --rts=1

Show all signals: *# pmshell –signals*

DSR=1 DTR=1 CTS=1 RTS=1 DCD=0

Read a line of text from the serial port: *# pmshell –getline*

Note: *For console servers running firmware v3.11.0 and above,* pmshell *contains a set of built-in key sequences to access the power menu, return to the serial port selection menu and more. Extra controls (key sequences) can be added to the built-in set of key sequences and configured per serial port. All ports can function the same or you can selectively add control sequences to ports. The controls can differ from port to port for the same function.*

For example, you can configure *pmshell* so that when you are using serial port 2, pressing *Ctrl+p* can take you straight to the power menu for that port.

The *pmshell* control commands are configured only via command line.

A helper script will configure a control command on a range of serial ports to eliminate the task of entering the configuration command for each port. You will still need to use this script once per control function (see below). There are six control functions.

pmshell control functions and their built in key sequences:

~b - Generate BREAK - send a break to the console

~h - View history - see the traffic logs for the port - must have port logging enabled

~p - Power menu - open the power menu for the port - port must be configured for an RPC

~m - Connect to port menu - go back to the serial port selection menu

~. - Exit pmshell - exit pmshell completely

~? - Show help message - shows the help message

Per Port Control Command Config Parameters:

config.ports.portX.ctrlcode.break	- Generate BREAK
config.ports.portX.ctrlcode.portlog	- View History
config.ports.portX.ctrlcode.power	- 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

The *pmshell* help message is NOT updated with the extra control command keys that may be configured. For example, to configure the *Ctrl+p* command 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.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 range of control codes 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 also works.

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 in the command line:

```
/etc/scripts/set-serial-control-codes 4 10 power 16
```

This sets the power menu control key to Ctrl+p (see the range of control codes below).

Note: *If you have not configured a particular serial port in the included range, configuration for that port will be skipped.*

Control Codes (Ctrl+a=1 ... Ctrl+z=26):

Ctrl+a = 1

Ctrl+b = 2

Ctrl+c = 3

Ctrl+d = 4

Ctrl+e = 5

Ctrl+f = 6

Ctrl+g = 7

Ctrl+h = 8

Ctrl+i = 9

Ctrl+j = 10

Ctrl+k = 11

Ctrl+l = 12

Ctrl+m = 13

Ctrl+n = 14

Ctrl+o = 15

Ctrl+p = 16

Ctrl+q = 17

Ctrl+r = 18

Ctrl+s = 19

Ctrl+t = 20

Ctrl+u = 21

Ctrl+v = 22

Ctrl+w = 23

Ctrl+x = 24

Ctrl+y = 25

Ctrl+z = 26

pmchat

The *pmchat* command acts similar to the standard *chat* command, but all serial port access is directed *via* the portmanager.

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 UNIX man pages:

http://techpubs.sgi.com/library/tpl/cgibin/getdoc.cgi?coll=linux&db=man&fname=/usr/share/catman/man8/chat.8.html

pmusers

The *pmusers* command is used to query the portmanager for active user sessions.

Example: To detect which users are currently active on which serial ports:

```
# pmusers
```

This command will output nothing if there are no active users currently connected to any ports, otherwise it will respond with a sorted list of usernames per active port:

```
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

Note: *With v3.11 firmware and later, the* pmusers *command is extended with the* --disconnect *option, which allows an* admin *user or* root *to disconnect console server sessions from the command line. The following connection types can be disconnected:*

telnet

SSH

Raw TCP

Unauth'ed Telnet

You cannot disconnect an *RFC2217* session.

If the *--disconnect* option is specified, the *pmusers* command goes into disconnect mode where you can specify the users with *-u*, the 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 overridden with the *--no-prompt* argument.

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 daemon

There is normally no need to stop and restart the daemon. To restart the daemon normally, just run the command:

```
# portmanager
```

Supported command line options are:

Force portmanager to run in the foreground:	*--nodaemon*
Set the level of debug logging:	*--loglevel={debug,info,warn,error,alert}*
Change which configuration file it uses:	*-c /etc/config/portmanager.conf*

Signals

Sending a SIGHUP signal to the portmanager will cause it to re-read its configuration file

15.2.2 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 the portmanager opens a port, it attempts to execute */etc/config/scripts/portXX.init* (where XX is the number of the port, *e.g.* 08). The script is run with STDIN and STDOUT both connected to the serial port.
- If the script cannot be executed, then portmanager will execute */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, the portmanager will attempt to execute */etc/config/scripts/portXX.alert* (where XX is the port number, e.g. 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, then 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>
```

15.3 Raw Access to Serial Ports

15.3.1 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 "stick" until that port is closed and opened again. So it is doubtful that people will want to use stty for more than initial debugging of the serial connection.

If you want to use stty to configure the 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 get lost when the portmanager opens the port. (the reason that portmanager sets things back to its config rather than using whatever is on the port, is 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).

15.3.2 Accessing the console/modem port

The console dial-in is handled by mgetty, with automatic PPP login extensions. *mgetty* is a smart *getty* replacement, designed to be used with 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), UUCP locking (so you can use the same device for dial-in and dial-out). *mgetty* provides very extensive logging facilities. All standard *mgetty* options are supported.

Modem initialization strings:

- To override the standard modem initialization string either use the Management Console (refer *Chapter 5*) or the command line config tool (refer *Dial-In Configuration 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
```

15.4 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 contained in configuration files:

/etc/config/ipfilter

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 current *iptables* configuration is erased
- If a customized IP-Filter script exists it is executed and no other actions are performed
- Standard policies are inserted which will drop all traffic not explicitly allowed to and through the system
- Rules are added which explicitly allow network traffic to access enabled services *e.g.* HTTP, SNMP etc
- Rules are added which explicitly allow traffic network traffic access to serial ports over enabled protocols *e.g.* Telnet, SSH and raw TCP

If the standard system firewall configuration is not adequate for your needs it can be bypassed safely by creating a file at **/etc/config/filter-custom** containing commands to build a specialized firewall. This firewall script will be run whenever the LAN interface is brought up (including initially) and will override any automated system firewall settings.

Below is a simple example of a custom script which creates a firewall using the iptables command. Only incoming connections from computers on a C-class network 192.168.10.0 will be accepted when this script is installed at */etc/config/filter-custom*. Note that when this script is called any preexisting chains and rules have been flushed from *iptables*:

```
#/bin/sh
# Set default policies to drop any incoming or routable traffic
# and blindly accept anything from the 192.168.10.0 network.
iptables –-policy FORWARD DROP
iptables –-policy INPUT DROP
iptables –-policy OUTPUT ACCEPT
# Allow responses to outbound connections back in.
iptables –-append INPUT \
        –-match state –-state ESTABLISHED,RELATED –-jump ACCEPT
# Explicitly accept any connections from computers on
# 192.168.10.0/24
iptables –-append INPUT –-source 192.168.10.0/24 –-jump ACCEPT
```

There's good documentation about using the iptables command at the Linux *netfilter* website *http://netfilter.org/documentation/index.html.* There are also many high-quality tutorials and HOWTOs available *via* the *netfilter* website, in particular peruse the tutorials listed on the *netfilter* HOWTO page.

15.5 SNMP Status Reporting and Traps

Console Servers can send traps/messages to multiple remote SNMP Network Managers on defined trigger events (as detailed in Chapter 7). Console Servers also contain an SNMP Service (snmpd) which can provide status information on demand. From the snmpd manual page:

> *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.*

15.5.1 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*.

OG-STATUS-MIB	This new MIB contains serial and connected device status information (for snmpstatusd & snmpalertd)
OG-STATUSv2-MIB	This new MIB contains extended status and alert
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)

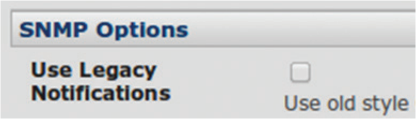
15.5.2 Check firewall rules

- Select **System: Services** and ensure the **SNMP daemon** box has been checked for the interface required

This will allow SNMP requests through the firewall for the specified interface.

15.5.3 Enable SNMP service

Note: *For firmware versions 3.10.2 and above, a 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 the SNMP menu to* ***Use Legacy Notifications*** *for the SNMP traps. In setting this option, the console server will send SNMP traps that are compatible with those sent from older firmware versions before new MIBS were added. This ensures that the firmware upgrade will not upset the existing SNMP management settings already in place.*



SNMP Options

Use Legacy Notifications □ Use old style

When upgrading from an old firmware version that does not support newer SNMP MIBs/traps (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, then the console server will be configured to use the new SNMP traps after the upgrade. Note: this will not have any effect until the SNMP service is turned on and an SNMP manager is configured.
- When starting up using the new firmware after a config erase, the console server will be configured to use the new SNMP traps.
- When upgrading from a firmware version that supports the new traps to a newer version that supports the new traps, the 'use legacy traps' setting should be kept the same – no checking SNMP service/manager configuration is needed.

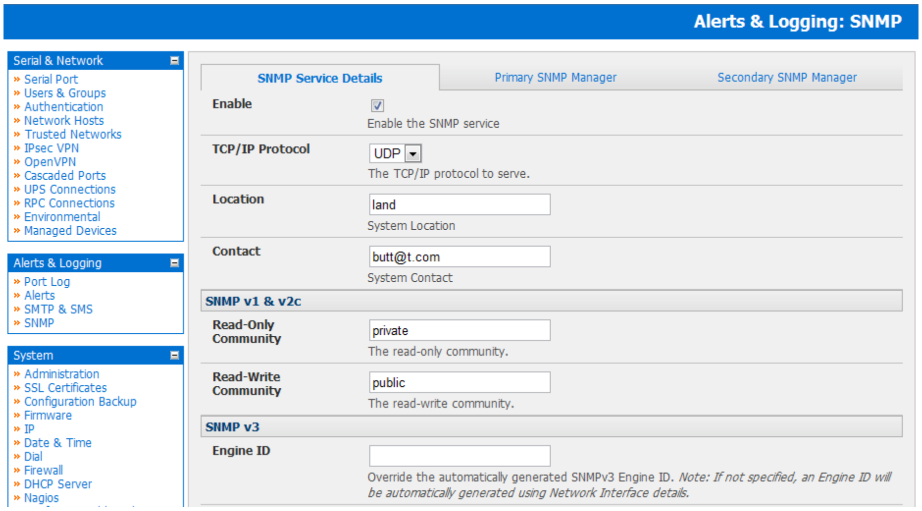
The Console Server supports different versions of SNMP including SNMPv1, SNMPv2c and SNMPv3.

SNMP, although an industry standard, 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 suggests 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
http://www.ietf.org/html.charters/snmpv3-charter.html.

- Select **Alerts & Logging: SNMP**



Alerts & Logging: SNMP

Enable ☑ Enable the SNMP service

TCP/IP Protocol UDP ▾ The TCP/IP protocol to serve.

Location land System Location

Contact butt@t.com System Contact

SNMP v1 & v2c

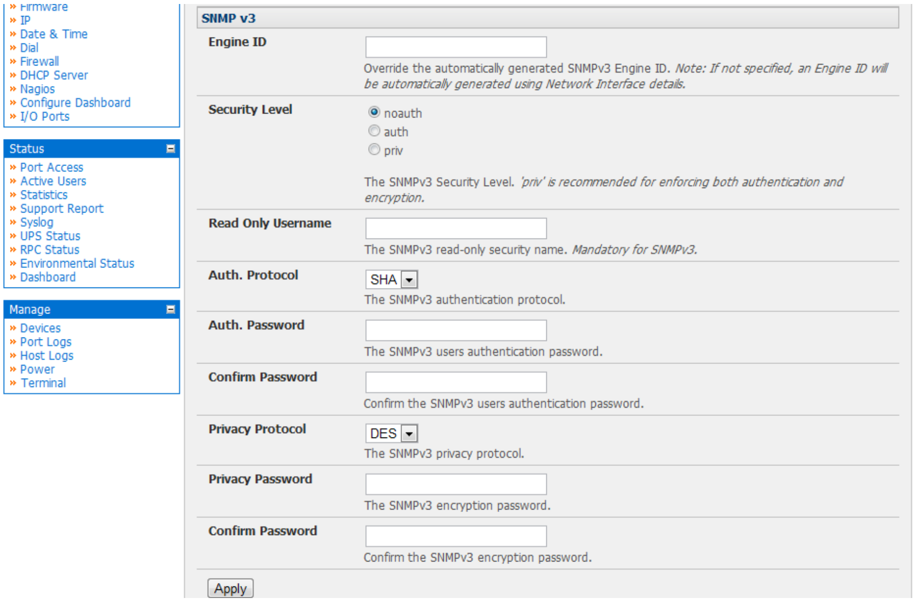
Read-Only Community private The read-only community.

Read-Write Community public The read-write community.

SNMP v3

Engine ID Override the automatically generated SNMPv3 Engine ID. *Note: If not specified, an Engine ID will be automatically generated using Network Interface details.*

- The **SNMP Service Details** tab is shown by default. The SNMP Service Details tab controls aspects of the SNMP Service including Security Level. It manages requests from external agents for status information.
- Check the **Enable the SNMP Service** box to start the SNMP Service. The Service 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 Console Server device is required.
- Complete the **Location** and **Contact** fields. The Location field should describe the physical location of the Console Server 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 Console Server 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.



SNMP v3

Engine ID Override the automatically generated SNMPv3 Engine ID. *Note: If not specified, an Engine ID will be automatically generated using Network Interface details.*

Security Level ● noauth ○ auth ○ priv The SNMPv3 Security Level. *'priv' is recommended for enforcing both authentication and encryption.*

Read Only Username The SNMPv3 read-only security name. *Mandatory for SNMPv3.*

Auth. Protocol SHA ▾ The SNMPv3 authentication protocol.

Auth. Password The SNMPv3 users authentication password.

Confirm Password Confirm the SNMPv3 users authentication password.

Privacy Protocol DES ▾ The SNMPv3 privacy protocol.

Privacy Password The SNMPv3 encryption password.

Confirm Password Confirm the SNMPv3 encryption password.

- Configure SNMP v3, if required. SNMP v3 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.
    - The **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 e.g. 0x01020304.
    - Specify the **Security Level**:

noauth No authentication or encryption is required. This is the minimum level of security.

auth Authentication will be required but encryption is not enforced. An authentication protocol (SHA or MD5) and password will be required.

priv Enforces the use of encryption. 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.

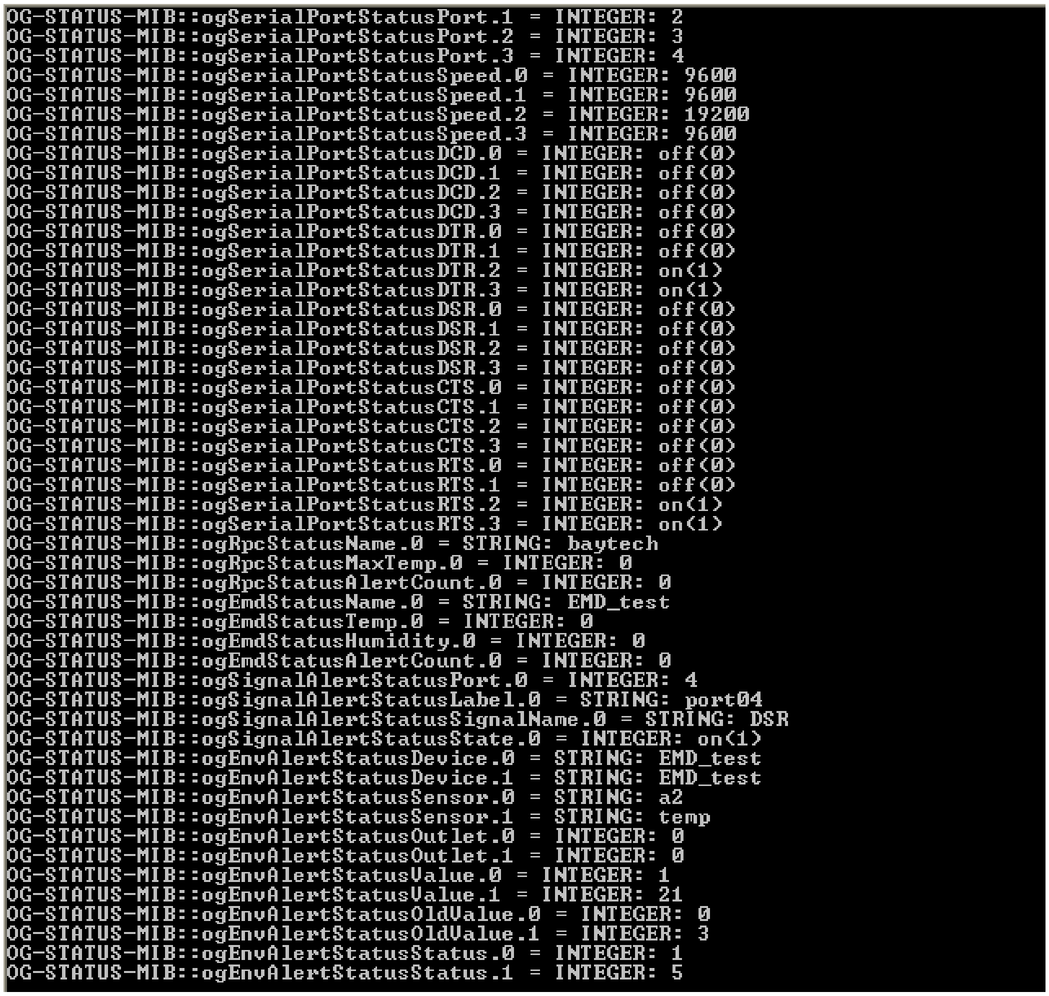
- o Complete the **Read Only Username**. Enter the read only security name. This field is mandatory and must be completed when configuring the Console Server for SNMPv3.
- o For a **Security Level** of **auth**, select the **Auth**. **Protocol (SHA** or **MD5)** and the **Auth**. **Password**. A password of at least 8 characters is required.
- o For a **Security Level** of **priv**, select the **Privacy Protocol** (**DES** or **AES**) and the **Privacy Password**. **AES** is recommended as it provides stronger privacy but requires more intense calculations. A password of at least 8 characters is required.

- Click **Apply**
- Setup serial ports and devices as per operational requirements such as UPS, RPC/PDU and EMD
- Copy the mibs from /etc/snmp/mibs on the Console Server product to a local directory using *scp* or *Winscp*. For example:

  *scp root@b095:/etc/snmp/mibs/**

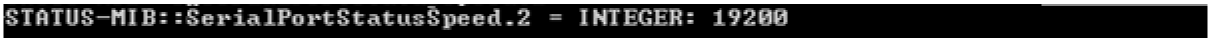
- Using the *snmpwalk* and *snmpget* commands, the status information can be retrieved from any console server. For example:

  snmpwalk -Oa -v1 -M .:/usr/share/snmp/mibs -c public b095 STATUS-MIB::ogStatus



```
OG-STATUS-MIB::ogSerialPortStatusPort.1 = INTEGER: 2
OG-STATUS-MIB::ogSerialPortStatusPort.2 = INTEGER: 3
OG-STATUS-MIB::ogSerialPortStatusPort.3 = INTEGER: 4
OG-STATUS-MIB::ogSerialPortStatusSpeed.0 = INTEGER: 9600
OG-STATUS-MIB::ogSerialPortStatusSpeed.1 = INTEGER: 9600
OG-STATUS-MIB::ogSerialPortStatusSpeed.2 = INTEGER: 19200
OG-STATUS-MIB::ogSerialPortStatusSpeed.3 = INTEGER: 9600
OG-STATUS-MIB::ogSerialPortStatusDCD.0 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusDCD.1 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusDCD.2 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusDCD.3 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusDTR.0 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusDTR.1 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusDTR.2 = INTEGER: on(1)
OG-STATUS-MIB::ogSerialPortStatusDTR.3 = INTEGER: on(1)
OG-STATUS-MIB::ogSerialPortStatusDSR.0 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusDSR.1 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusDSR.2 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusDSR.3 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusCTS.0 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusCTS.1 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusCTS.2 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusCTS.3 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusRTS.0 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusRTS.1 = INTEGER: off(0)
OG-STATUS-MIB::ogSerialPortStatusRTS.2 = INTEGER: on(1)
OG-STATUS-MIB::ogSerialPortStatusRTS.3 = INTEGER: on(1)
OG-STATUS-MIB::ogRpcStatusName.0 = STRING: baytech
OG-STATUS-MIB::ogRpcStatusMaxTemp.0 = INTEGER: 0
OG-STATUS-MIB::ogRpcStatusAlertCount.0 = INTEGER: 0
OG-STATUS-MIB::ogEmdStatusName.0 = STRING: EMD_test
OG-STATUS-MIB::ogEmdStatusTemp.0 = INTEGER: 0
OG-STATUS-MIB::ogEmdStatusHumidity.0 = INTEGER: 0
OG-STATUS-MIB::ogEmdStatusAlertCount.0 = INTEGER: 0
OG-STATUS-MIB::ogSignalAlertStatusPort.0 = INTEGER: 4
OG-STATUS-MIB::ogSignalAlertStatusLabel.0 = STRING: port04
OG-STATUS-MIB::ogSignalAlertStatusSignalName.0 = STRING: DSR
OG-STATUS-MIB::ogSignalAlertStatusState.0 = INTEGER: on(1)
OG-STATUS-MIB::ogEnvAlertStatusDevice.0 = STRING: EMD_test
OG-STATUS-MIB::ogEnvAlertStatusDevice.1 = STRING: EMD_test
OG-STATUS-MIB::ogEnvAlertStatusSensor.0 = STRING: a2
OG-STATUS-MIB::ogEnvAlertStatusSensor.1 = STRING: temp
OG-STATUS-MIB::ogEnvAlertStatusOutlet.0 = INTEGER: 0
OG-STATUS-MIB::ogEnvAlertStatusOutlet.1 = INTEGER: 0
OG-STATUS-MIB::ogEnvAlertStatusValue.0 = INTEGER: 1
OG-STATUS-MIB::ogEnvAlertStatusValue.1 = INTEGER: 21
OG-STATUS-MIB::ogEnvAlertStatusOldValue.0 = INTEGER: 0
OG-STATUS-MIB::ogEnvAlertStatusOldValue.1 = INTEGER: 3
OG-STATUS-MIB::ogEnvAlertStatusStatus.0 = INTEGER: 1
OG-STATUS-MIB::ogEnvAlertStatusStatus.1 = INTEGER: 5
```

*snmpget -Oa -v1 -M .:/usr/share/snmp/mibs -c public b095 OG-STATUSMIB::
ogSerialPortStatusSpeed.2*



```
STATUS-MIB::SerialPortStatusSpeed.2 = INTEGER: 19200
```

noauth

snmpwalk -Oa –v3 –l noAuthNoPriv –u readonlyusername -M .:/usr/share/snmp/mibs b095 STATUS-MIB::Status

auth

snmpwalk -Oa –v3 –l authNoPriv –u readonlyusername –a SHA –A “authpassword” -M .:/usr/share/snmp/mibs b095 STATUS-MIB::ogStatus

priv

snmpwalk -Oa –v3 –l authNoPriv –u readonlyusername –a SHA –A “authpassword” –x DES –X “privpassword” -M .:/usr/share/snmp/mibs b095 STATUS-MIB::ogStatus

-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 may be used to explore the enterprise MIB structure.

15.5.4 /etc/config/snmpd.conf

The *net-snmpd* is an extensible SNMP which includes built-in support for a wide range of MIB information modules, and can be extended using dynamically loaded modules, external scripts and commands. *snmpd* when enabled should run with a default configuration. Its behavior can be customized via the options in */etc/config/snmpd.conf.*

Note: If the SNMP Service is enabled through the Web Based Management Console this configuration file will be overidden and you will lose any customization.

Changing standard system information such as system contact, name and location can be achieved by editing */etc/config/snmpd.conf* file and locating the following lines:

```
sysdescr       "tripplite"
syscontact     root <root@localhost>(configure /etc/default/snmpd.conf)
sysname        Not defined (edit /etc/default/snmpd.conf)
syslocation    Not defined (edit /etc/default/snmpd.conf)
```

Simply change the values of *sysdescr*, *syscontact*, *sysname* and *syslocation* to the desired settings and restart *snmpd*.

The *snmpd.conf* provides is extremely powerful and too flexible to completely cover here. The configuration file itself is commented extensively and good documentation is available at the net-s*nmp* website http://www.net-snmp.org, specifically:

Man Page:	http://www.net-snmp.org/docs/man/snmpd.conf.html
FAQ:	http://www.net-snmp.org/docs/FAQ.html
Net-SNMPD Tutorial:	http://www.net-snmp.org/tutorial/tutorial-5/demon/snmpd.html

15.5.5 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 (refer Chapter 7) 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. Refer back to the Management Console UI or user documentation for descriptions of each field.

To 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 DNS name.

To set the Manager Trap Port field

```
   config --set=”config.system.snmp.trapport3=162”
```

.. replacing 162 with the TCP/UDP port number

To set the SNMP Manager Protocol field:

```
  config --set="config.system.snmp.protocol3=UDP" or
  config --set="config.system.snmp.protocol3=TCP"
```

To set the SNMP Manager Version field:

```
  config --set="config.system.snmp.version3=3"
```

To set the SNMP Manager v1 & v2c community field:

```
  config --set="config.system.snmp.community3=public"
```

To set the SNMP Manager v3 Engine ID field:

```
  config –set="config.system.snmp.engineid3=0x8000000001020304"
```

.. replacing 0x8000000001020304 with the hex Engine-ID

To 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"
```

To set the SNMP Manager v3 Username field:

```
  config --set="config.system.snmp.username3=username"
```

To 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, e.g. config.system.snmp.protocol3, config.system.snmp.address3, etc

15.6 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

15.6.1 SSH Overview

Popular TCP/IP applications such as *telnet, rlogin, ftp*, and others transmit their passwords unencrypted. Doing this across public 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. http://www.openssh.com/ The only changes in the 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. e.g.*
    - o */etc/config/sshd_config* instead of */etc/sshd_config*
    - o */etc/config/ssh_config* instead of */etc/ssh_config*
    - o */etc/config/users/<username>/.ssh/* instead of */home/<username>/.ssh/*

15.6.2 Generating Public Keys (Linux)

To generate new SSH key pairs use the Linux *ssh-keygen* command. This will produce an RSA or DSA public/private key pair and you will be prompted for a path to store the two key files e.g. *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_[rsa|dsa]):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_[rsa|dsa].
Your public key has been saved in /home/user/.ssh/id_[rsa|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:

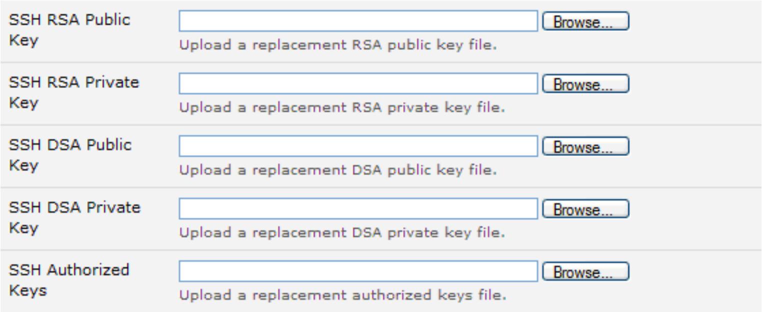
```
$ mkdir keys
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):  /home/user/keys/control_room
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/keys/control_room
Your public key has been saved in /home/user/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
$
```

You must ensure there is no password associated with the keys. If there is a password, then the Console Server devices will have no way to supply it as runtime.

Full documentation for the *ssh-keygen* command can be found at *http://www.openbsd.org/cgi-bin/man.cgi?query=ssh-keygen*

15.6.3 Installing the SSH Public/Private Keys (Clustering)

For Console Servers the keys can be simply uploaded through the web interface, on the **System: Administration** page. This enables you to upload stored RSA or DSA Public Key pairs to the Master and apply the Authorized key to the slave and is described in Chapter 4. Once complete you then proceed to Fingerprinting as described below.



SSH RSA Public Key	Upload a replacement RSA public key file.
SSH RSA Private Key	Upload a replacement RSA private key file.
SSH DSA Public Key	Upload a replacement DSA public key file.
SSH DSA Private Key	Upload a replacement DSA private key file.
SSH Authorized Keys	Upload a replacement authorized keys file.

15.6.4 Installing SSH Public Key Authentication (Linux)

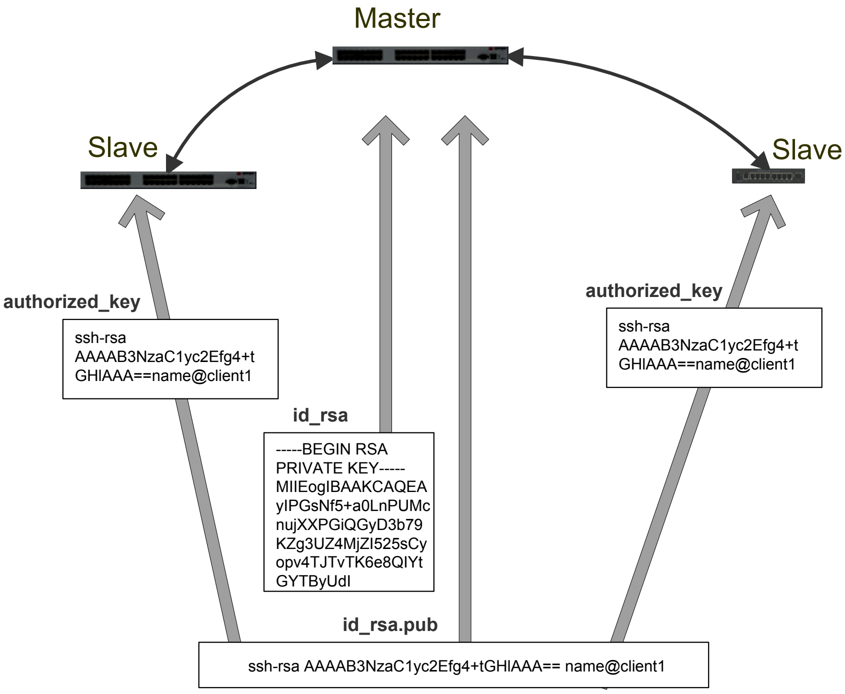
Alternately the public key can be installed on the unit remotely from the linux host with the *scp* utility as follows.

Assuming the user on the Management Console is called "fred"; the IP address of the Console Server is 192.168.0.1 (default); and the public key is on the *linux/unix* computer in *~/.ssh/id_dsa.pub*. Execute the following command on the linux/unix computer:

```
scp ~/.ssh/id_dsa.pub \
root@192.168.0.1:/etc/config/users/fred/.ssh/authorized_keys
```

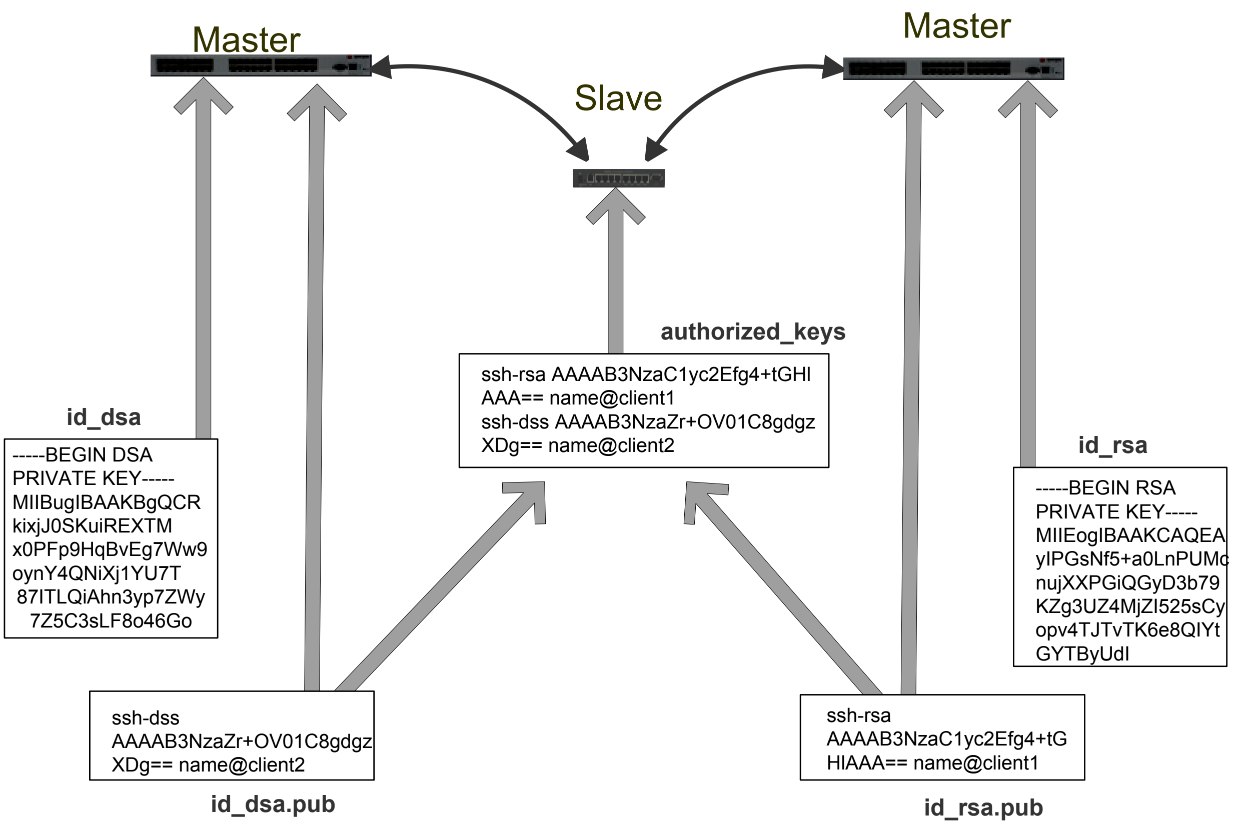
The authorized_keys file on the Console Server needs to be owned by "fred", so login to the Management Console as **root** and type:

```
chown fred /etc/config/users/fred/.ssh/authorized_keys
```



If the Console Server device selected to be the server will only have one client device, then 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 server, then the *authorized_keys* file will contain a copy 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:

$ ls /home/user/keys control_room control_room.pub plant_entrance plant_entrance.pub $ cat /home/user/keys/control_room.pub /home/user/keys/plant_entrance.pub > /home/user/keys/authorized_keys_bridge_server



```
ssh-rsa AAAAB3NzaC1yc2Efg4+tGHI
AAA== name@client1
ssh-dss AAAAB3NzaZr+OV01C8gdgz
XDg== name@client2
```

```
-----BEGIN DSA
PRIVATE KEY-----
MIIBugIBAAKBgQCR
kixjJ0SKuiREXTM
x0PFp9HqBvEg7Ww9
oynY4QNiXj1YU7T
 87ITLQiAhn3yp7ZWy
  7Z5C3sLF8o46Go
```

```
-----BEGIN RSA
PRIVATE KEY-----
MIIEogIBAAKCAQEA
yIPGsNf5+a0LnPUMc
nujXXPGiQGyD3b79
KZg3UZ4MjZI525sCy
opv4TJTvTK6e8QIYt
GYTByUdI
```

```
ssh-dss
AAAAB3NzaZr+OV01C8gdgz
XDg== name@client2
```

```
ssh-rsa
AAAAB3NzaC1yc2Efg4+tG
HIAAA== name@client1
```

More documentation on OpenSSH can be found at:

http://openssh.org/portable.html
http://www.openbsd.org/cgi-bin/man.cgi?query=ssh&sektion=1
http://www openbsd.org/cgi-bin/man.cgi?query=sshd.

15.6.5 Generating public/private keys for SSH (Windows)

This section describes how to generate and configure SSH keys using Windows.

First create a new user from the Console Server Management Console (the following example uses a user called "testuser") making sure it is a member of the "users" group.

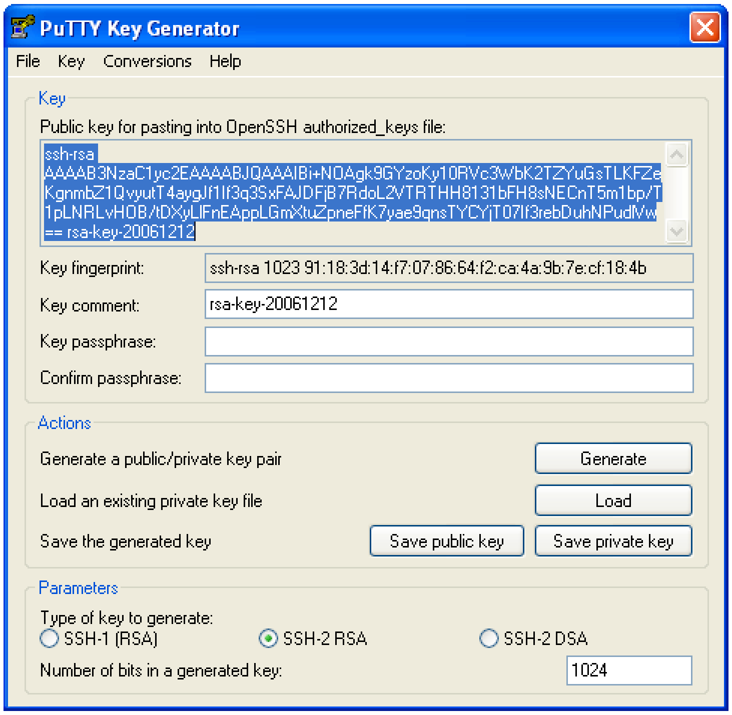
If you do not already have a public/private key pair you can generate them now using *ssh-keygen, PuTTYgen* or a similar tool:

PuTTYgen: *http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html*
OpenSSH: *http://www.openssh.org/*
OpenSSH (Windows): *http://sshwindows.sourceforge.net/download/*

For example using PuTTYgen, make sure you have a recent version of the *puttygen.exe* (available from *http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html*) Make sure you have a recent version of WinSCP (available from *http://winscp.net/eng/download.php*)

To generate a SSH key using PuTTY *http://sourceforge.net/docs/F02/#clients:*

- Execute the PUTTYGEN.EXE program
- Select the desired key type SSH2 DSA (you may use RSA or DSA) within the *Parameters* section
- It is important that you leave the passphrase field blank
- Click on the *Generate* button
- Follow the instruction to move the mouse over the blank area of the program in order to create random data used by PUTTYGEN to generate secure keys. Key generation will occur once PUTTYGEN has collected sufficient random data
- Create a new file " authorized_keys " (with notepad) and copy your public key data from the "Public key for pasting into



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAIBi+NOAgk9GYzoKy10RVc3WbK2TZYuGsTLKFZe
KgnmbZ1QvyutT4aygJf1If3q3SxFAJDFjB7RdoL2VTRTHH8131bFH8sNECnT5m1bp/T
1pLNRLvHOB/tDXyLIFnEAppLGmXtuZpneFfK7yae9qnsTYCYjT07If3rebDuhNPudMw
== rsa-key-20061212
```

Key fingerprint: ssh-rsa 1023 91:18:3d:14:f7:07:86:64:f2:ca:4a:9b:7e:cf:18:4b

Key comment: rsa-key-20061212

Key passphrase:

Confirm passphrase:

Generate a public/private key pair

Load an existing private key file

Save the generated key

Type of key to generate:
○ SSH-1 (RSA) ● SSH-2 RSA ○ SSH-2 DSA

Number of bits in a generated key: 1024

OpenSSH authorized_keys file" section of the PuTTY Key Generator, and paste the key data to the "authorized_keys" file. Make sure there is only one line of text in this file

- Use WinSCP to copy this "authorized_keys" file into the user's home directory: eg. /etc/config/users/testuser/.ssh/authorized_keys of the Console Server which will be the SSH server. You will need to make sure this file is in the correct format with the correct permissions with the following commands:

  ```
  # dos2unix  \
  /etc/config/users/testuser/.ssh/authorized_keys && chown testuser \
  /etc/config/users/testuser/.ssh/authorized_keys
  ```

- Using WinSCP copy the attached sshd_config over */etc/config/sshd_config* on the server (Makes sure public key authentication is enabled)

- Test the Public Key by logging in as "testuser" Test the Public Key by logging in as "testuser" to the client Console Server device and typing (you should not need to enter anything): # 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 clients */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.

15.6.6 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 remhost
    The authenticity of host 'remhost (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)?
```

At this stage, answer yes to accept the key. You should get the following message:

Warning: Permanently added 'remhost,192.168.0.1' (RSA) to the list of known hosts.

You may be prompted for a password, but there is no need to log in - you have received the fingerprint and can 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 (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*

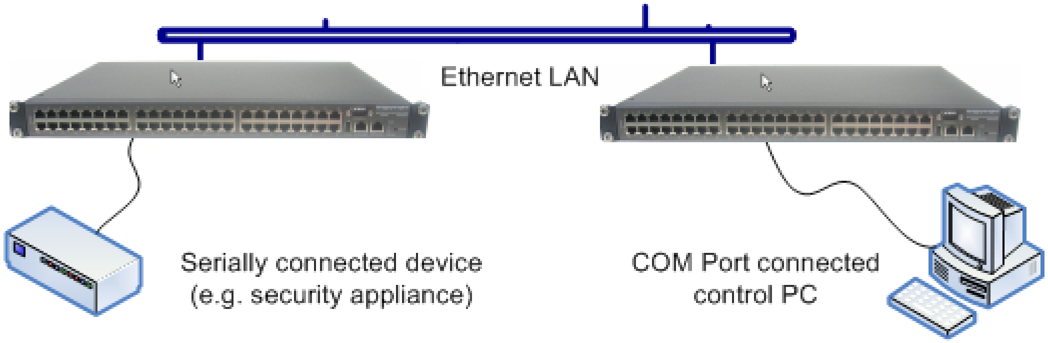
RSA host key for *remhost* has changed and you have requested strict checking.

Host key verification failed.

If the host key has been legitimately changed, it can be removed from the ~/.ssh/known_hosts file and the new fingerprint added. If it has not changed, this indicates a serious problem that should be investigated immediately.

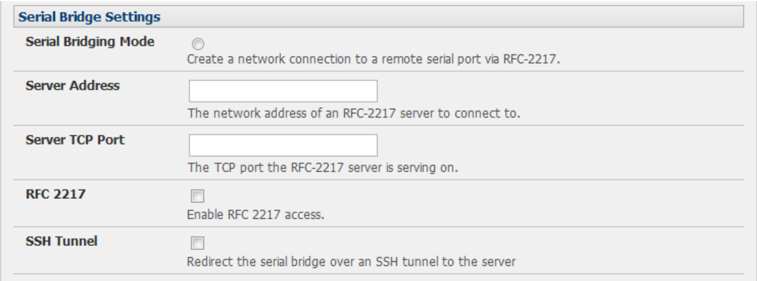
15.6.7 SSH tunneled serial bridging

You have the option to apply SSH tunneling when two Black Box console servers are configured for serial bridging.



As detailed in *Chapter 4*, 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**



Serial Bridge Settings

Serial Bridging Mode	○ Create a network connection to a remote serial port via RFC-2217.
Server Address	 The network address of an RFC-2217 server to connect to.
Server TCP Port	 The TCP port the RFC-2217 server is serving on.
RFC 2217	□ Enable RFC 2217 access.
SSH Tunnel	□ Redirect the serial bridge over an SSH tunnel to the server

Next you will need to set up SSH keys for each end of the tunnel and upload these keys to the Server and Client console servers.

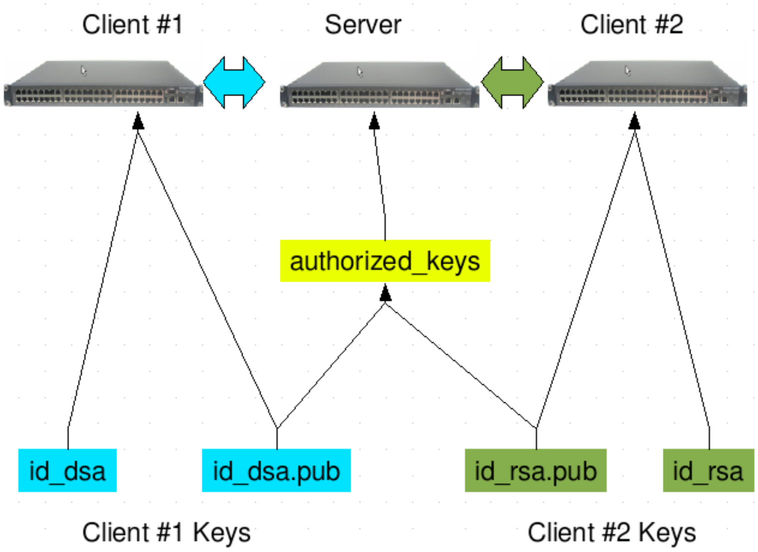
Client Keys:

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*.

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 using OpenBSD's OpenSSH suite, we use the ssh-keygen program:

```
$ ssh-keygen -t [rsa|dsa]
Generating public/private [rsa|dsa] key pair.
Enter file in which to save the key (/home/user/.ssh/id_[rsa|dsa]):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_[rsa|dsa].
Your public key has been saved in /home/user/.ssh/id_[rsa|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 (/home/user/.ssh/id_rsa):  /home/user/keys/control_room
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/keys/control_room
Your public key has been saved in /home/user/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
$
```

You should ensure there is no password associated with the keys. If there is a password, then the Console Servers will have no way to supply it as runtime.

Authorized Keys:

If the Console Server selected to be the server will only have one client device, then 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 server, then the *authorized_keys* file will contain a copy 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:

```
$ ls /home/user/keys
control_room control_room.pub plant_entrance plant_entrance.pub
$ cat /home/user/keys/control_room.pub
/home/user/keys/plant_entrance.pub >
/home/user/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) go in the correct spots, and that the public and private keys are in the correct spot.

15.6.8 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 (i.e. public key authentication).

- To use public key authentication with SDT Connector, first you must 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 – as described in the earlier section.
- Next, add the private part of your SSH key pair (this file is typically named *id_rsa* or *id_dsa*) to SDT Connector client. Click **Edit: Preferences: Private Keys: Add**, locate the private key file and 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, and 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 (e.g. Putty, OpenSSH) and the host's SSH server for public key authentication.

15.7 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 excellent SSLeay library developed by Eric A. Young and Tim J. Hudson. The OpenSSL toolkit is licensed under an Apache-style licence, 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 'http' in order to have secure browser access to the GUI management console across insecure networks.

More documentation on OpenSSL is available from:

http://www.openssl.org/docs/apps/openssl.html

http://www.openssl.org/docs/HOWTO/certificates.txt

15.8 HTTPS

The Management Console can be served using HTTPS by running the webserver via *sslwrap*. The server can be launched on request using inetd.

The HTTP server provided is a slightly modified version of the *fnord-httpd* from *http://www.fefe.de/fnord/*

The SSL implementation is provided by the *sslwrap* application compiled with OpenSSL support. More detailed documentation can be found at *http://www.rickk.com/sslwrap/*

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.

15.8.1 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
```

15.8.2 Generating a self-signed certificate with OpenSSL

This example shows how to use OpenSSL to create a self-signed certificate. OpenSSL is available for most Linux distributions via the default package management mechanism. (Windows users can check *http://www.openssl.org/related/binaries.html*)

To create a 1024 bit RSA key and a self-signed certificate issue the following *openssl* 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. When you have entered everything, the certificate will be created in a file called *ssl_cert.pem*.

15.8.3 Installing the key and certificate

The recommended method for copying files securely to the Console Server unit is with an SCP (Secure Copying Protocol) client. The *scp* utility is distributed with OpenSSH for most Unix distributions while 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*

More detailed documentation on the PSCP can be found:
http://the.earth.li/~sgtatham/putty/0.58/htmldoc/Chapter5.html#pscp

15.8.4 Launching the HTTPS Server

Note that the easiest way to enable the HTTPS server is from the web Management Console. Simply click the appropriate checkbox in **Network: Services: HTTPS Server** and the HTTPS server will be activated (assuming the *ssl_key.pem* & *ssl_cert.pem* files exist in the */etc/config directory*).

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 and signal *inetd* of the configuration change.

```
kill -HUP `cat /var/run/inetd.pid`
```

The HTTPS server should be accessible from a web client at a URL similar to this: *https://<common name of unit>*

More detailed documentation about the *openssl* utility can be found at the website: *http://www.openssl.org/*

15.9 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 8. These RPCs are controlled using the open source *NUT* and *PowerMan* tools and the *pmpower* utility.

15.9.1 PowerMan

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. Target hostnames are mapped to plugs on RPC devices in powerman.conf
powerman - power on/off nodes

Synopsis
powerman [-option] [targets]
pm [-option] [targets]

Options
-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, powermand 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 are 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 are 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 refer *http://linux.die.net/man/1/powerman*. Also refer powermand (*http://linux.die.net/man/1/powermand*) documentation and powerman.conf (*http://linux.die.net/man/5/powerman.conf*)

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 (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:

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]"

15.9.2 pmpower

The *pmpower* command is a high-level tool for manipulating remote, preconfigured power devices connected to the Console Servers either via a serial or network connection.

pmpower [-?h] [-l device | -r host] [-o outlet] [-u username] [-p password] action

-?/-h	This help message.
-l	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 located at IP address 192.168.1.100 to OFF (where username is 'root' and 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.

15.9.3 Adding new RPC devices

There are two simple paths to adding support for new RPC devices.

The first is to have scripts to support the particular RPC included in the open source *PowerMan* project (*http://sourceforge.net/projects/powerman*). The *PowerMan* device specifications are unusual 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*. Once the new RPC support has been built into the *PowerMan*, we will include the updated *PowerMan* build in a subsequent firmware release.

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.

The user can add their own 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, log in 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, its 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.

15.10 IPMItool

The Console Server includes the *ipmitool* utility for managing and configuring devices that support the Intelligent Platform Management Interface (IPMI) version 1.5 and version 2.0 specifications.

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 the 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.

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

-a Prompt for the remote server password.

-A *<authtype>* Specify an authentication type to use during IPMIv1.5 *lan* session activation. Supported types are NONE, PASSWORD, MD5, or OEM.

-c Present output in CSV (comma separated variable) format. This is not available with all commands.

-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 an 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 and visible in the usage help output.

-L *<privlvl>* Force session privilege level. Can be CALLBACK, USER, OPERATOR, 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 that you should not enable IPMI for remote access without setting a password, and that the 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. Passwords longer than 16 characters will be truncated.

For IPMI v2.0, the maximum password length is 20 characters; longer passwords are truncated.

COMMANDS

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.

ipmitool help

Commands:

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

ipmitool chassis help

Chassis Commands: status, power, identify, policy, restart_cause, poh, bootdev

ipmitool chassis power help

chassis power Commands: status, on, off, cycle, reset, diag, soft

You will find more details on *ipmitools* at http://*ipmitool*.sourceforge.net/manpage.html

15.11 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. However, the Master does not provide a fully consolidated view, e.g. *Status: Active Users*. It 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 parse each port log file line by line. Each time it sees 'LOGIN: *username*', it adds the 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 is also possible to run this as a CGI script on the B092-016. In this case, the remote/USB logged port logs files are in: */var/run/portmanager/logdir* (or they are in */var/log*). Otherwise you can run the 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* and click Apply

To run the CGI script on the Console Server:

- Login to the B092-016
- Run: *mount -o remount,rw /dev/hda1 /*
- Copy the script to */home/httpd/cgi-bin/*
- Run: *mount -o remount,ro /dev/hda1 /*
- Browse to: http://192.168.0.1/cgi-bin/yourscript.cgi where 192.168.0.1 is the IP address of the Console Server and *yourscript.cgi* is the name of the script

There is a useful tutorial on creating a bash script CGI at http://www.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*

15.12 SMS Server Tools

Console Servers include the *SMS Server Tools software* which provides an SMS Gateway which 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 the WAP / MMS capable mobile phone.

The program can be run as a 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, this limit is easily changeable).

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). AT command set is supported. Devices can be connected with serial port, infrared or USB.

For more information, refer to *http://smstools3.kekekasvi.com*

15.13 Multicast

By default, Console Servers have Multicasting enabled. Multicasting provides products with the ability to simultaneously transmit information from a single device to a select group of hosts.

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.

Chapter 16: Thin Client

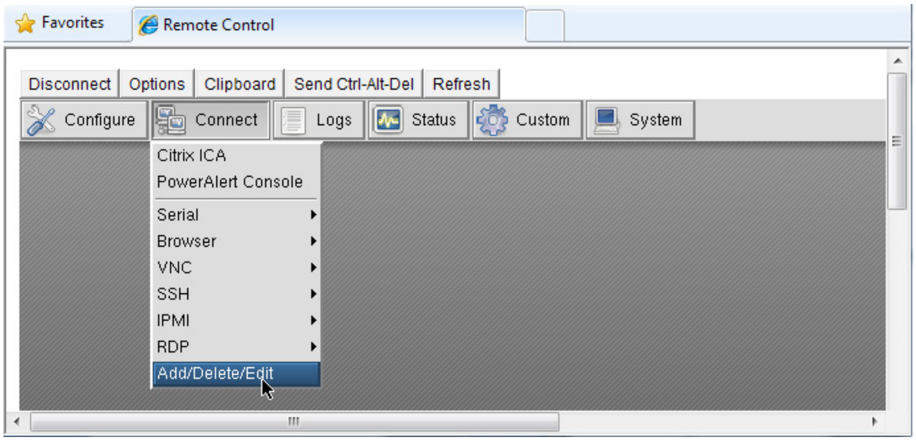
The B092-016 has a selection of management clients (Firefox browser, SSH, Telnet, VNC viewer, ICA, RDP) embedded as well as the Tripp Lite PowerAlert software. With these, the B092-016 provides rackside control of computers, networking, telecom, power and other managed devices via serial, USB or IP over the LAN.

This chapter provides instructions on configuring the thin clients and using them locally and remotely. The thin clients can be controlled from the rack side using a direct monitor/keyboard/mouse connected to the B092-016 or remotely using a VNC connection from the remote user to the B092-016.

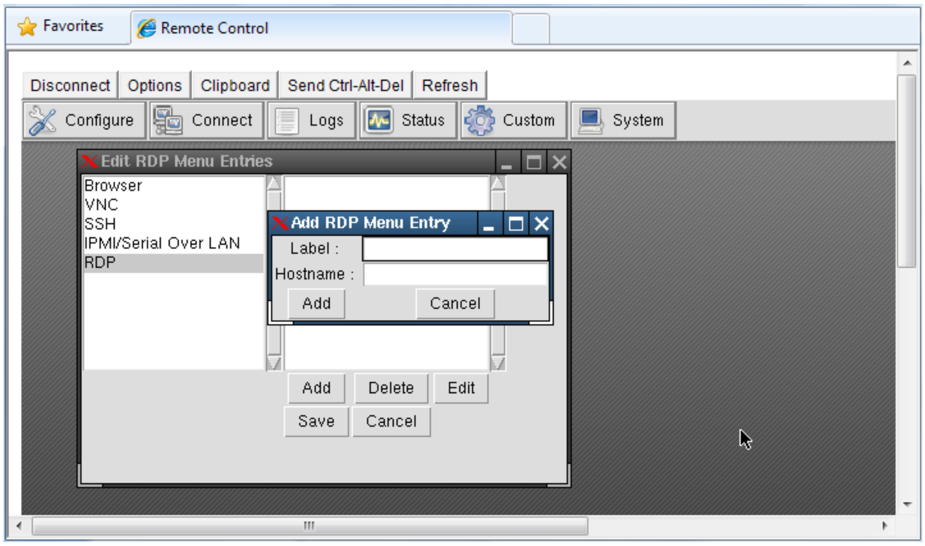
16.1 Local Client Service Connections

These client connections first need to be configured:

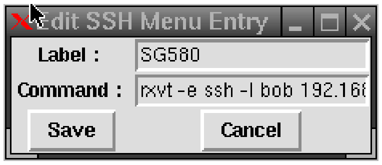
- Select **Connect: Add/Delete/Edit** on the control panel



- Then select a *Connect* client (such as RDP) and click **Add** to configure the Host connection for that client service



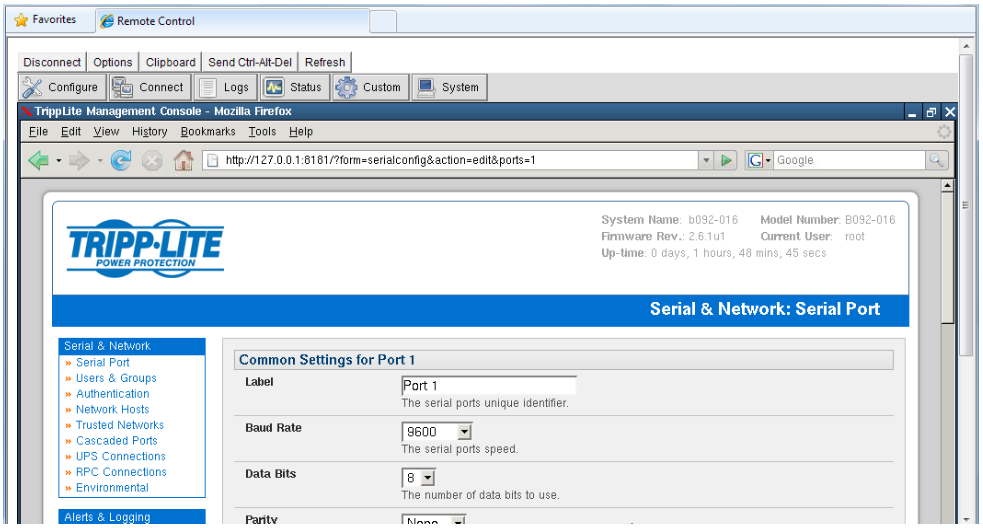
- For each new Host you add, you will be asked to enter a **Label** (enter a descriptive name) and a **Hostname** (enter the **IP Address** or **DNS Name** of the new network connected Host) and possibly a **Username** (enter the name you will use to log in to the Host)
- Once a Host has been added, you can select **Edit** and update the commands that will be executed in connecting the service to the existing Host



Label : SG580

Command : rxvt -e ssh -l bob 192.16

- The sixteen serial ports are pre-configured by default in Console Server mode for the B096-016 / B096-032 / B096-048 Console Server Management Switch or in UPS (PowerAlert) mode for the B092-016 Console Server with PowerAlert product. To change these settings, select Configure, which will load the local Firefox browser and run the Management Console. You can then reconfigure the serial ports as detailed in Chapter 4



http://127.0.0.1:8181/?form=serialconfig&action=edit&ports=1

System Name: b092-016 Model Number: B092-016
Firmware Rev.: 2.6.1u1 Current User: root
Up-time: 0 days, 1 hours, 48 mins, 45 secs

Serial & Network: Serial Port

Common Settings for Port 1

Label: Port 1
The serial ports unique identifier.

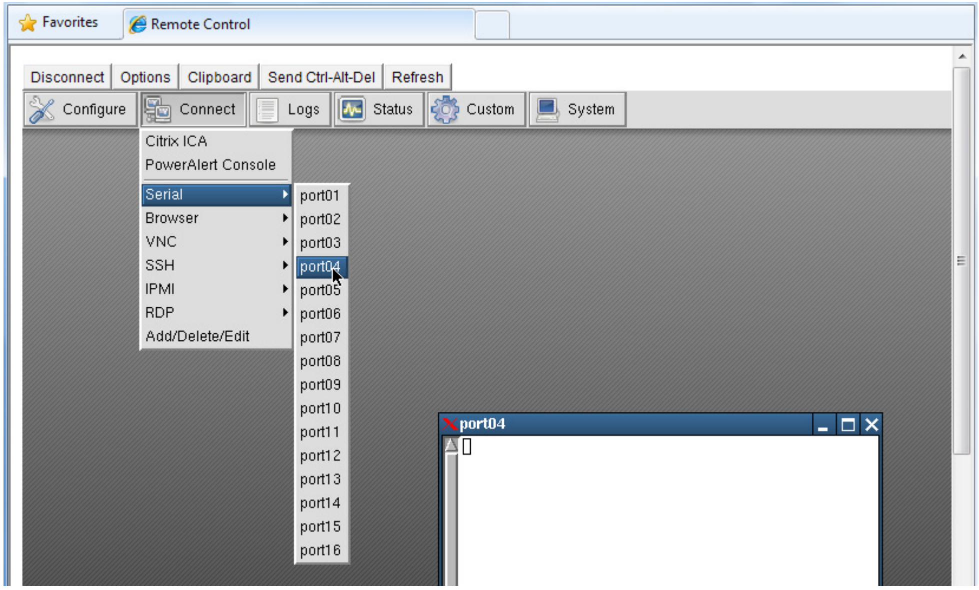
Baud Rate: 9600
The serial ports speed.

Data Bits: 8
The number of data bits to use.

Parity: None

16.1.1 Connect: Serial Terminal

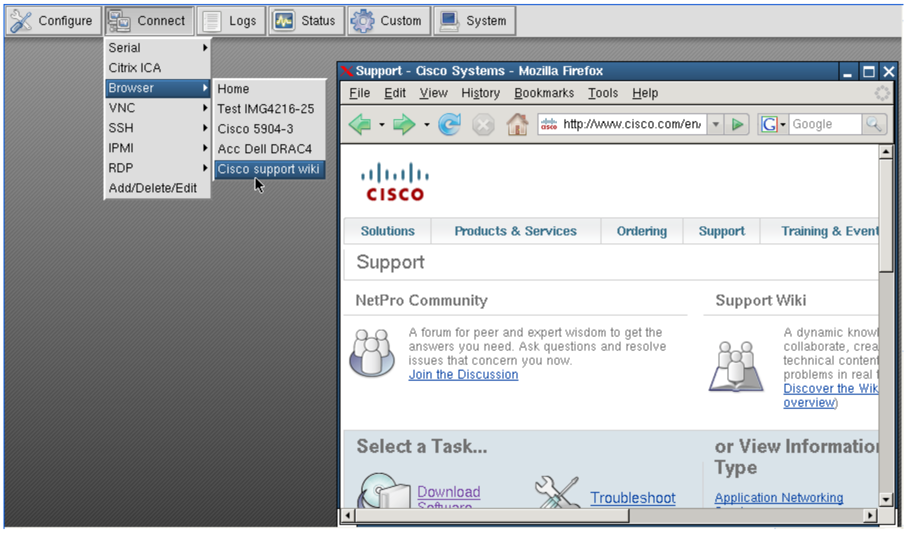
- Select **Connect: Serial** on the control panel and click on the desired serial port. A window will be created with a connection to the device on the selected serial port:



The embedded terminal emulator uses *rxvt* (a color vt102 terminal emulator). You can find more details on configuration options in http://www.rxvt.org/manual.html

16.1.2 Connect: Browser

- Select **Connect: Browser** on the control panel and click on the Host/web site you have configured to be accessed using the browser. Sites can be internal or external.



The B092-016 provides a powerful Mozilla Firefox browser with a licensed Sun Java JRE

version 2.0.0.11

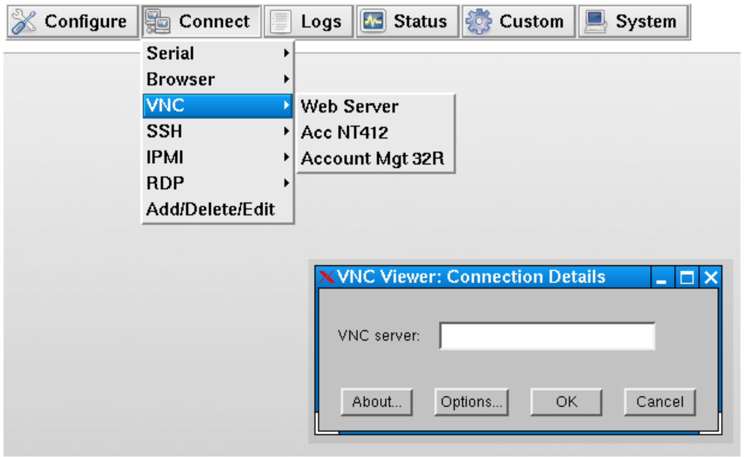
©1998-2007 Contributors. All Rights Reserved. Firefox and the Firefox logos are trademarks of the Mozilla Foundation. All rights reserved. Some trademark rights used under license from The Charlton Company.

Mozilla/5.0 (X11; U; Linux i586; en-US; rv:1.8.1.11) Gecko/20080730 Firefox/2.0.0.11

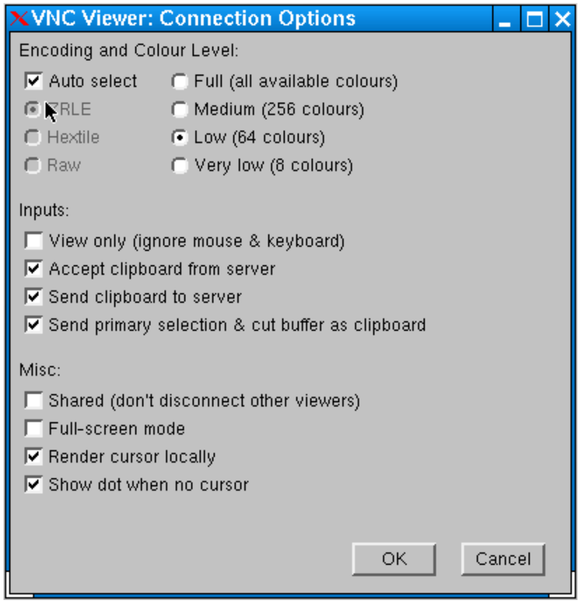
Java and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries

16.1.3 Connect: VNC

- Select **Connect: VNC** on the control panel and click on the VNC server Host to be accessed
- The VNC Viewer client in your B092-016 will be started and a VNC connection window to the selected server will be opened



- If the *HostName* was left blank when the VNC server connection was configured, then the VNC Viewer will start with a request for the VNC server.
- Selecting **Options** at this stage enables you to configure the VNC Viewer
- Alternately, you can select *Options* by right-clicking on the VNC Viewer task Bar icon



VNC Viewer: Connection Options

Encoding and Colour Level:

☑ Auto select
● ZRLE
○ Hextile
○ Raw
○ Full (all available colours)
○ Medium (256 colours)
● Low (64 colours)
○ Very low (8 colours)

Inputs:

☐ View only (ignore mouse & keyboard)
☑ Accept clipboard from server
☑ Send clipboard to server
☑ Send primary selection & cut buffer as clipboard

Misc:

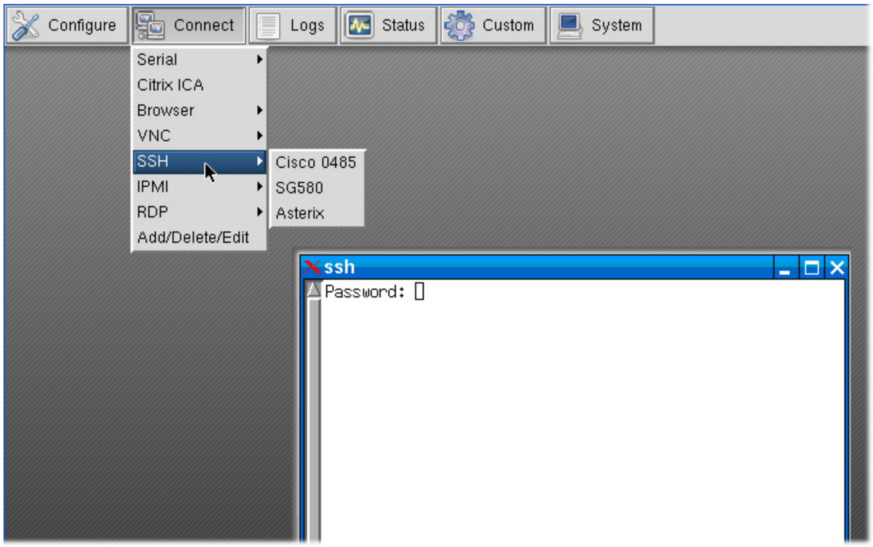
☐ Shared (don't disconnect other viewers)
☐ Full-screen mode
☑ Render cursor locally
☑ Show dot when no cursor

You can find more details on configuration options in *http://www.realvnc.com/products/free/4.1/man/vncviewer.html*

16.1.4 Connect: SSH

SSH is typically used to log into a remote machine and execute commands.

- Select **Connect: SSH** on the control panel and click on the Host to be accessed
- An SSH connection window will be opened. Enter the SSH login password and you will be securely connected to the selected Host



```
Password: []
```

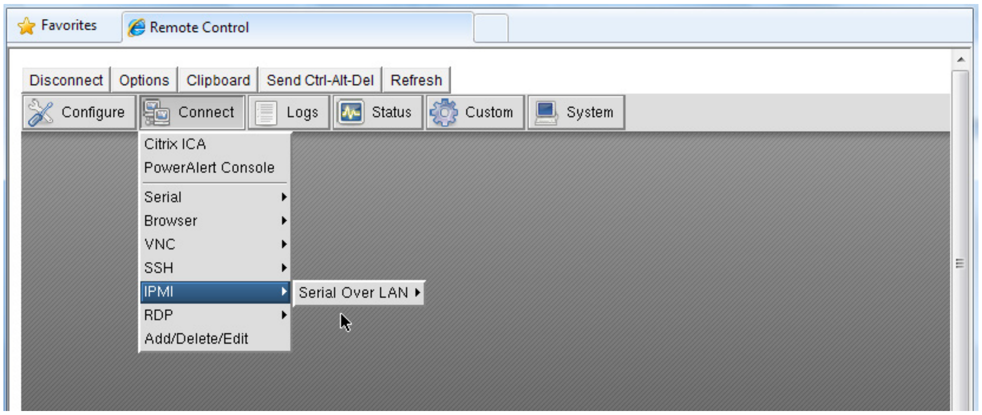
The B092-016 SSH connection uses OpenSSH (http://www.openssh.com/) and the terminal connection is presented using rxvt (*ou RXVT*). You can find more details on configuration options in http://www.rxvt.org/manual.html

16.1.5 Connect: IPMI

The B092-016 control panel provides a number of IPMI tools for managing service processors or Baseboard Management Controllers (BMCs). These IPMI controls are built on the *ipmitools* program. Find more details on configuration options in http://ipmitool.sourceforge.net/manpage.html

The ipmitool program provides a simple command-line interface to the BMCs and features the ability to read the sensor data repository (SDR), display the contents of the System Event Log (SEL), read and set LAN configuration parameters, and perform remote chassis power control. The B092-016 Management Console also has additional tools for controlling power units with IPMI interfaces (refer to *Chapter 8*).

- Select **Connect: IPMI** on the control panel and select the **Serial over LAN** connection to be accessed

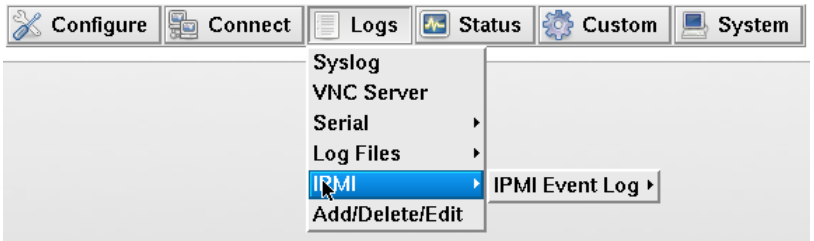


This will launch a Serial-Over-LAN session by running:

# ipmitool -I lanplus -H hostname -U username -P password sol activate

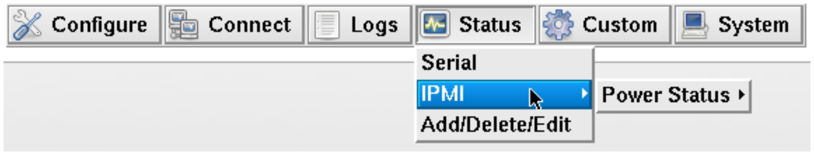
The resulting serial character connection is presented in an *rxvt* (*ouR XVT*) window. Also the Serial-Over-LAN feature is only applicable to IPMI2.0 devices.

- Select **Logs: IPMI** on the control panel and select the IPMI Event Log to be viewed



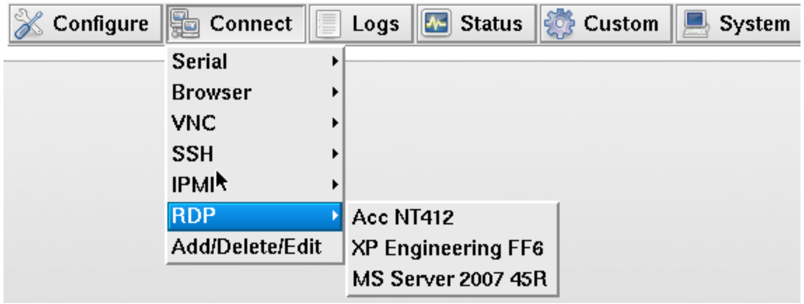
This will retrieve the selected IPMI event log by running:

# ipmitool -I lanplus -H hostname -U username -P password sel info

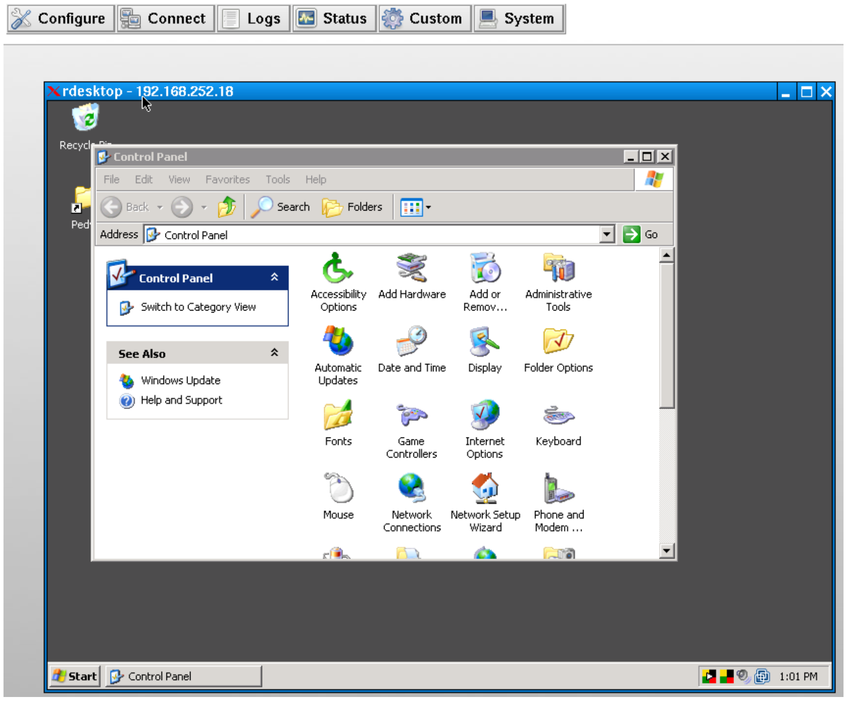


16.1.6 Connect: Remote Desktop (RDP)

- Select **Connect: RDP** on the control panel and click on the Windows computer to be accessed



- The *rdesktop* program in your B092-016 will be started, an RDP connection to the Remote Desktop server in the selected computer will be opened, the *rdesktop* window will appear on your B092-016 screen and you will be prompted for a password. (If the selected computer does not have RDP access enabled, then the *rdesktop* window will not appear.)



You can use Add/Delete/Edit to customize the rdesktop client (e.g. to include login username passwords). The command line protocol is:

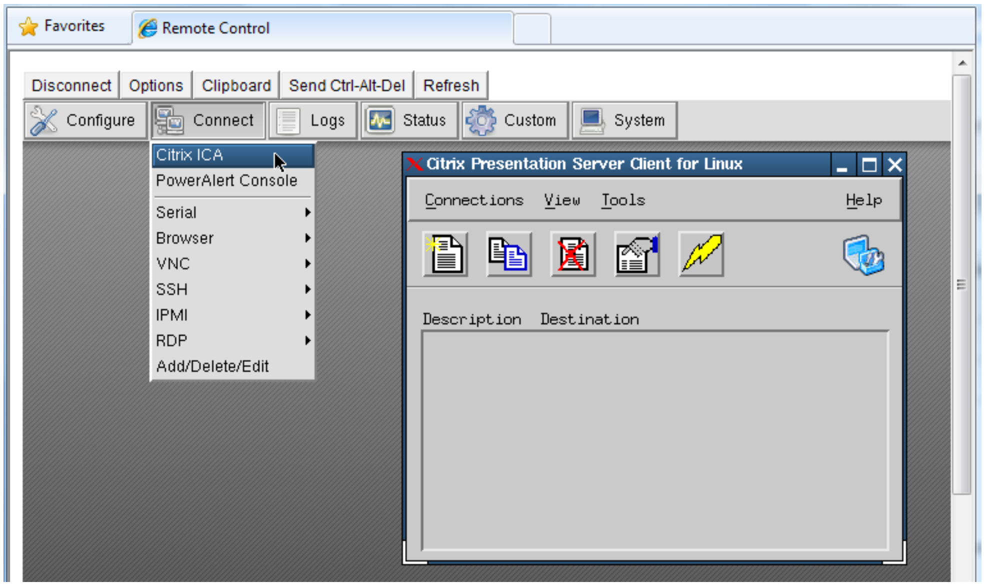
rdesktop -u windows-user-id -p windows-password -g 1200x950 ms-windows-terminal-server-host-name

option	Description
-a	Color depth: 8, 16, 24
-r	Device redirection. i.e. Redirect sound on remote machine to local device i.e. -0 -r sound (MS/Windows 2003)
-g	Geometry: widthxheight or 70% screen percentage.
-p	Use -p - to receive password prompt.

Further information on *rdesktop* can be found at http://www.rdesktop.org/

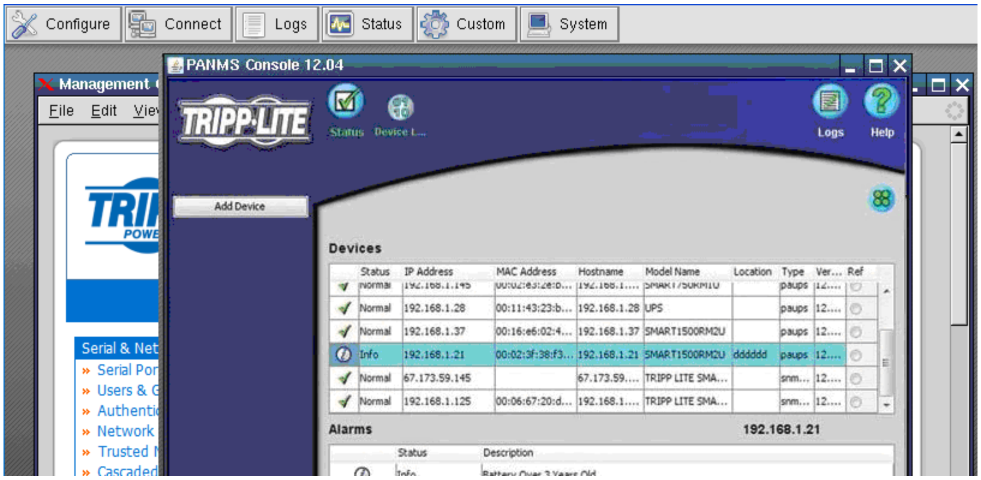
16.1.7 Connect: Citrix ICA

- Select **Connect: Citrix ICA** on the control panel and click on the Citrix server to be accessed



16.1.8 Connect: PowerAlert

- Select **Connect: PowerAlert** on the control panel. The PowerAlert software will be launched.



Devices

	Status	IP Address	MAC Address	Hostname	Model Name	Location	Type	Ver...	Ref
✓	Normal	192.168.1.145	00:02:e3:2e:b...	192.168.1....	SMART750RM1U		paups	12....	○
✓	Normal	192.168.1.28	00:11:43:23:b...	192.168.1.28	UPS		paups	12....	○
✓	Normal	192.168.1.37	00:16:e6:02:4...	192.168.1.37	SMART1500RM2U		paups	12....	○
ⓘ	Info	192.168.1.21	00:02:3f:38:f3...	192.168.1.21	SMART1500RM2U	dddddd	paups	12....	○
✓	Normal	67.173.59.145		67.173.59....	TRIPP LITE SMA...		snm...	12....	○
✓	Normal	192.168.1.125	00:06:67:20:d...	192.168.1....	TRIPP LITE SMA...		snm...	12....	○

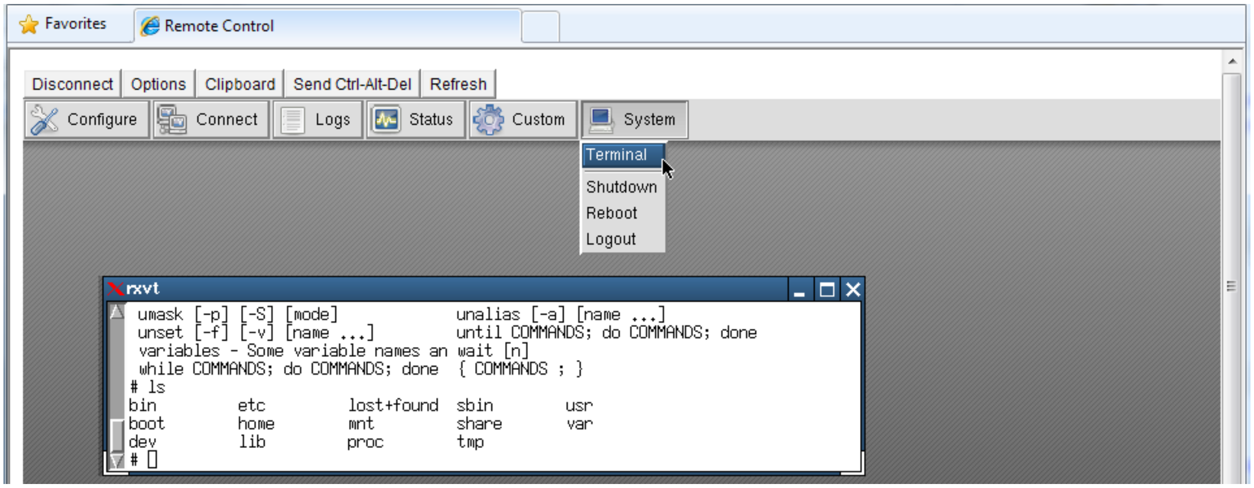
Alarms 192.168.1.21

	Status	Description
ⓘ	Info	Battery Over 3 Years Old

16.2 Advanced Control Panel

16.2.1 System: Terminal

Selecting System: Terminal on the control panel logs you in at the command line to the B092-016 Linux kernel. As detailed in Chapters 14 and 15, this enables you to configure and customize your B092-016 using the config and portmanager commands or general Linux commands.



```
 umask [-p] [-S] [mode]          unalias [-a] [name ...]
 unset [-f] [-v] [name ...]      until COMMANDS; do COMMANDS; done
 variables - Some variable names an wait [n]
 while COMMANDS; do COMMANDS; done  { COMMANDS ; }
# ls
bin         etc         lost+found  sbin        usr
boot        home        mnt         share       var
dev         lib         proc        tmp
# 
```

16.2.2 System: Shutdown / Reboot

Clicking **System: Shutdown** on the control panel will shut down the B092-016 system. You will need to cycle the power to reactivate the B092-01.

Similarly, by clicking **System: Reboot**, you will initiate a *soft* reset. With a soft reset, the B092-016 reboots with all settings such as the assigned network IP address, preserved. However a *soft* reset disconnects all Users and ends any SSH sessions that had been established.

A *soft* reset will also occur when you switch OFF power from the B092-016, and then switch the power back ON. However, if you cycle the power while the unit is writing to flash, you could corrupt or lose data, so the software **Shutdown** or **Reboot** from the control panel is the safer option.

16.2.3 System: Logout

Clicking **System: Logout** closes the local user log in session (and removes the control panel). However, this does not logout remote users who may be logged into the B092-016 Console Server, or accessing attached devices using SSH tunneling.

16.2.4 Custom

The Custom button on the control panel enables you to customize your B092-016 by adding buttons to the control panel that execute bash and other Linux commands you specify.

16.2.5 Status

These menu items give the user a snapshot of the serial port and IPMI device status.

16.2.6 Logs

These menu items give the user an audit log of B092-016 activity.

16.3 Remote Control

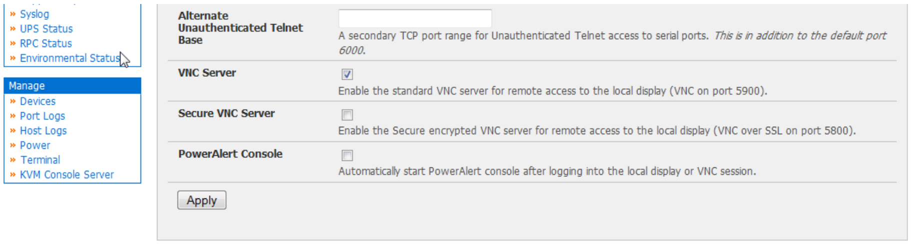
You can access the B092-016 locally via a directly connected keyboard, monitor and mouse (or KVM switch). If the B092-016 is connected to a KVMoIP infrastructure, then this may also provide you with some remote access to the B092-016 local consoles (RDP, Telnet, VNC, ICA, JRE etc).

The B092-016 also hosts an embedded VNC server that enables you to remotely monitor and control the thin client software (RDP, Telnet, VNC, ICA etc) that is running in the B092-016 itself.

Note: You can still run management client software (RDP etc) on the remote computer and use SDT to securely connect the client directly to the managed devices that are serially or network attached to the B092-016. This is useful when running proprietary applications (such as Dell OpenManage) or Windows applications (such as VMware VDI client) on a remote management computer which is be used to manage a DRAC service processor or VMware virtual device on a remote server.

Each B092-016 gateway has an internal VNC server enabling remote administrators to oversee local activity, and giving them the option to access and control all the devices themselves. To activate the VNC server in the B092-016:

- Select the **System: Services** option in the *Management Console* menu then check **VNC Server** or **Secure VNC Server**



Alternate Unauthenticated Telnet Base
A secondary TCP port range for Unauthenticated Telnet access to serial ports. *This is in addition to the default port 6000.*

VNC Server ☑
Enable the standard VNC server for remote access to the local display (VNC on port 5900).

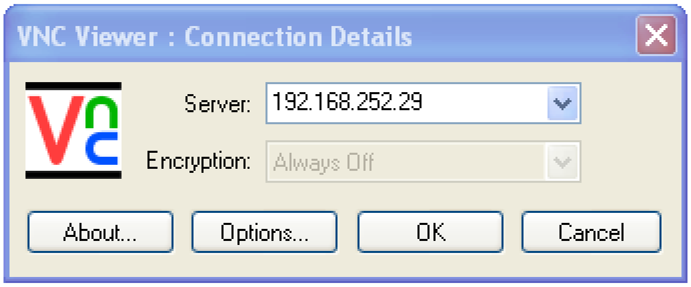
Secure VNC Server □
Enable the Secure encrypted VNC server for remote access to the local display (VNC over SSL on port 5800).

PowerAlert Console □
Automatically start PowerAlert console after logging into the local display or VNC session.

- Click **Manage: KVM Console Server** then **Launch Standard VNC Remote Control** and your browser will automatically download and run a Java VNC applet client
- Log in as *root* (or some other configured B092-016 username) and as a remote Administrator you can then connect to the VNC server in the B092-016 and gain remote access to (and monitor and take control of) the B092-016 local display

You can find more details on configuration options for the B092-016 *realvnc* server in
http://www.realvnc.com/products/free/4.1/man/vncserver.html

Note: You can also run a VNC client application such as RealVNC, TightVNC or UltraVNC directly on a remote computer and configure it with the B092-016's IP address to connect to the B092-016 VNC server



Server: 192.168.252.29
Encryption: Always Off

Appendix A: Hardware Specification

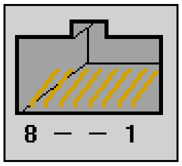
FEATURE	VALUE
Dimensions	B096-016 / B096-032 / B096-048: 17 x 12 x 1.75 in (43.2 x 31.3. x 4.5 cm) B092-016: 17 x 6.7 x 1.75 in (44 x 17 x 4.5 cm) B095-004 / B095-003: 4.1x3.4x1.1 in (10.3 x 8.7 x 2.8 cm) B094-008-2E-M-F / B094-008-2E-V: 6.5 x 4 x 1.4 in (16.6 x 10.2 x 2.8 cm)
Weight	B096-016 / B096-032 / B096-048: 11.8 lbs (5.4 kg) B092-016: 8.5 lb (3.9 kg) B095-004 / B095-003: 2.2 lbs (1.0 kg) B094-008-2E-M-F / B094-008-2E-V: 1.8 kg (4 lbs)
Ambient operating temperature	41°F to 122°F (5°C to 50°C)
Non-operating storage temperature	-20°F to +140°F (-30°C to +60°C)
Humidity	5% to 90%
Power	Refer to Chapter 2

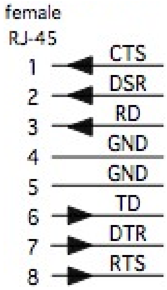
Appendix B: Serial Port Connectivity

Pinout standards exist for both DB9 and DB25 connectors. However, there are not pinout standards for serial connectivity using RJ45 connectors. Many Console Servers and serially managed servers/ router/ switches/ PSUs have adopted their own unique pinout; so custom connectors and cables may be required to interconnect your Console Server. In an endeavor to create some move to standardization, Tripp Lite Console Server products all use the same RJ45 pinout convention as adopted by Cisco, SUN and others.

Serial Port Pinout

The 16/48 RJ45 connectors on the B092-016 Console Server with PowerAlert, and the B096-048/032/016 Console Server Management Switch have the following pinout:





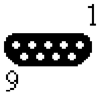
PIN	SIGNAL	DEFINITION	DIRECTION
1	CTS	Clear To Send	Input
2	DSR	Data Set Ready	Input
3	RXD	Receive Data	Input
4	GND	Signal Ground	NA
5	GND	Signal Ground	NA
6	TXD	Transmit Data	Output
7	DTR	Data Terminal Ready	Output
8	RTS	Request To Send	Output

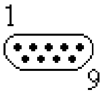
The LOCAL (console/modem) port on the Console Server uses a standard DB9 connector as tabled below:

SIGNAL	DB9 Pin	DEFINITION
TXD	3	Transmitted Data
RXD	2	Received Data
RTS	7	Request To Send
CTS	8	Clear To Send
DSR	6	Data Set Ready
GND	5	Signal Ground
CD	1	Received Line Signal Detector
DTR	4	Data Terminal Ready
RI	9	Ring Indicator

FEMALE MALE

9 pin DB9



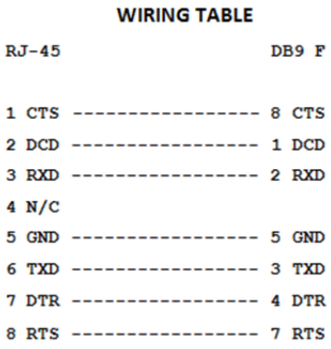


Serial Port Connectivity

Connectors included in Console Server

All products:

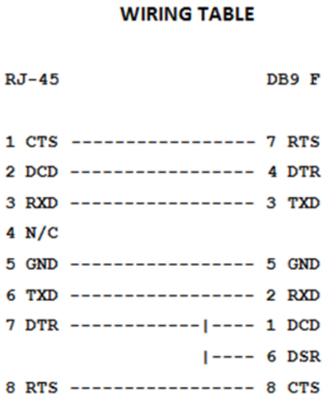
DB9F-RJ45S straight connector



WIRING TABLE

```
RJ-45                       DB9 F

1 CTS ----------------- 8 CTS
2 DCD ----------------- 1 DCD
3 RXD ----------------- 2 RXD
4 N/C
5 GND ----------------- 5 GND
6 TXD ----------------- 3 TXD
7 DTR ----------------- 4 DTR
8 RTS ----------------- 7 RTS
```

DB9F-RJ45S cross-over connector



WIRING TABLE

```
RJ-45                       DB9 F

1 CTS ----------------- 7 RTS
2 DCD ----------------- 4 DTR
3 RXD ----------------- 3 TXD
4 N/C
5 GND ----------------- 5 GND
6 TXD ----------------- 2 RXD
7 DTR ------------|---- 1 DCD
                  |---- 6 DSR
8 RTS ----------------- 8 CTS
```

Appendix C: End User License Agreements

READ BEFORE USING THE ACCOMPANYING SOFTWARE

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE USING THE ACCOMPANYING SOFTWARE, THE USE OF WHICH IS LICENSED FOR USE ONLY AS SET FORTH BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, DO NOT USE THE SOFTWARE. IF YOU USE ANY PART OF THE SOFTWARE, SUCH USE WILL INDICATE THAT YOU ACCEPT THESE TERMS.

You have acquired a product that includes Tripp Lite (“Tripp Lite”) proprietary software and/or proprietary software licensed to Tripp Lite. This Tripp Lite End User License Agreement (“EULA”) is a legal agreement between you (either an individual or a single entity) and Tripp Lite for the installed software product of Tripp Lite origin, as well as associated media, printed materials, and “online” or electronic documentation (“Software”). By installing, copying, downloading, accessing, or otherwise using the Software, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, Tripp Lite is not willing to license the Software to you. In such event, do not use or install the Software. If you have purchased the Software, promptly return the Software and all accompanying materials with proof of purchase for a refund.

Products with separate end user license agreements that may be provided along with the Software are licensed to you under the terms of those separate end user license agreements.

LICENSE GRANT. Subject to the terms and conditions of this EULA, Tripp Lite grants you a nonexclusive right and license to install and use the Software on a single CPU, provided that, (1) you may not rent, lease, sell, sublicense or lend the Software; (2) you may not reverse engineer, decompile, disassemble or modify the Software, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation; and (3) you may not transfer rights under this EULA unless such transfer is part of a permanent sale or transfer of the Product, you transfer at the same time all copies of the Software to the same party or destroy such materials not transferred, and the recipient agrees to this EULA.

No license is granted in any of the Software’s proprietary source code. This license does not grant you any rights to patents, copyright, trade secrets, trademarks or any other rights with respect to the Software.

You may make a reasonable number of copies of the electronic documentation accompanying the Software for each Software license you acquire, provided that, you must reproduce and include all copyright notices and any other proprietary rights notices appearing on the electronic documentation. Tripp Lite reserves all rights not expressly granted herein.

INTELLECTUAL PROPERTY RIGHTS. The Software is protected by copyright laws, international copyright treaties, and other intellectual property laws and treaties. Tripp Lite and its suppliers retain all ownership of, and intellectual property rights in (including copyright), the Software components and all copies thereof, provided however, that (1) certain components of the Software, including SDT Connector, are components licensed under the GNU General Public License Version 2, which Tripp Lite supports, and (2) the SDT Connector includes code from JSch, a pure Java implementation of SSH2 which is licensed under BSD style license. Copies of these licenses are detailed below and Tripp Lite will provide source code for any of the components of the Software licensed under the GNU General Public License upon request.

EXPORT RESTRICTIONS. You agree that you will not export or re-export the Software, any part thereof, or any process or service that is the direct product of the Software in violation of any applicable laws or regulations of the United States or the country in which you obtained them.

U.S. GOVERNMENT RESTRICTED RIGHTS. The Software and related documentation are provided with Restricted Rights. Use, duplication, or disclosure by the Government is subject to restrictions set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software – Restricted Rights at 48 C.F.R. 52.227-19, as applicable, or any successor regulations.

TERM AND TERMINATION. This EULA is effective until terminated. The EULA terminates immediately if you fail to comply with any term or condition. In such an event, you must destroy all copies of the Software. You may also terminate this EULA at any time by destroying the Software.

GOVERNING LAW AND ATTORNEY'S FEES. This EULA is governed by the laws of the State of Utah, USA, excluding its conflict of law rules. You agree that the United Nations Convention on Contracts for the International Sale of Goods is hereby excluded in its entirety and does not apply to this EULA. If you acquired this Software in a country outside of the United States, that country's laws may apply. In any action or suit to enforce any right or remedy under this EULA or to interpret any provision of this EULA, the prevailing party will be entitled to recover its costs, including reasonable attorneys' fees.

ENTIRE AGREEMENT. This EULA constitutes the entire agreement between you and Tripp Lite with respect to the Software, and supersedes all other agreements or representations, whether written or oral. The terms of this EULA can only be modified by express written consent of both parties. If any part of this EULA is held to be unenforceable as written, it will be enforced to the maximum extent allowed by applicable law, and will not affect the enforceability of any other part.

Should you have any questions concerning this EULA, or if you desire to contact Tripp Lite for any reason, please contact the Tripp Lite representative serving your company.

THE FOLLOWING DISCLAIMER OF WARRANTY AND LIMITATION OF LIABILITY IS INCORPORATED INTO THIS EULA BY REFERENCE. THE SOFTWARE IS NOT FAULT TOLERANT. YOU HAVE INDEPENDENTLY DETERMINED HOW TO USE THE SOFTWARE IN THE DEVICE, AND TRIPPLITE HAS RELIED UPON YOU TO CONDUCT SUFFICIENT TESTING TO DETERMINE THAT THE SOFTWARE IS SUITABLE FOR SUCH USE.

LIMITED WARRANTY Tripp Lite warrants the media containing the Software for a period of ninety (90) days from the date of original purchase from Tripp Lite or its authorized retailer. Proof of date of purchase will be required. Any updates to the Software provided by Tripp Lite (which may be provided by Tripp Lite at its sole discretion) shall be governed by the terms of this EULA. In the event the product fails to perform as warranted, Tripp Lite's sole obligation shall be, at Tripp Lite's discretion, to refund the purchase price paid by you for the Software on the defective media, or to replace the Software on new media. Tripp Lite makes no warranty or representation that its Software will meet your requirements, will work in combination with any hardware or application software products provided by third parties, that the operation of the software products will be uninterrupted or error free, or that all defects in the Software will be corrected.

TRIPP LITE DISCLAIMS ANY AND ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OTHER THAN AS STATED HEREIN, THE ENTIRE RISK AS TO SATISFACTORY QUALITY, PERFORMANCE, ACCURACY, AND EFFORT IS WITH YOU. ALSO, THERE IS NO WARRANTY AGAINST INTERFERENCE WITH YOUR ENJOYMENT OF THE SOFTWARE OR AGAINST INFRINGEMENT. IF YOU HAVE RECEIVED ANY WARRANTIES REGARDING THE DEVICE OR THE SOFTWARE, THOSE WARRANTIES DO NOT ORIGINATE FROM, AND ARE NOT BINDING ON, TRIPP LITE.

NO LIABILITY FOR CERTAIN DAMAGES. EXCEPT AS PROHIBITED BY LAW, TRIPP LITE SHALL HAVE NO LIABILITY FOR COSTS, LOSS, DAMAGES OR LOST OPPORTUNITY OF ANY TYPE WHATSOEVER, INCLUDING BUT NOT LIMITED TO, LOST OR ANTICIPATED PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, EXEMPLARY SPECIAL OR CONSEQUENTIAL DAMAGES, WHETHER UNDER CONTRACT, TORT, WARRANTY OR OTHERWISE ARISING FROM OR IN CONNECTION WITH THIS EULA OR THE USE OR PERFORMANCE OF THE SOFTWARE. IN NO EVENT SHALL TRIPPLITE BE LIABLE FOR ANY AMOUNT IN EXCESS OF THE LICENSE FEE PAID TO TRIPPLITE UNDER THIS EULA. SOME STATES AND COUNTRIES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION MAY NOT APPLY TO YOU.

JSch License

SDT Connector includes code from JSch, a pure Java implementation of SSH2. JSch is licensed under BSD style license and it is:

Copyright (c) 2002, 2003, 2004 Atsuhiko Yamanaka, JCraft, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SDT Connector License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

SUN Java License

(B092-016 Console Server with PowerAlert product only)

1. Java Technology Restrictions. Licensee shall not create, modify, change the behavior of, or authorize licensees of Licensee to create, modify, or change the behavior of, classes, interfaces, or subpackages that are in any way identified as "java", "javax", "sun" or similar convention as specified by Sun in any naming convention designation. In the event that Licensee creates an additional API(s) which: (a) extends the functionality of a Java Environment; and (b) is exposed to third party software developers for the purpose of developing additional software which invokes such additional API, Licensee must promptly publish broadly an accurate specification for such API for free use by all developers.

2. Trademarks and Logos. This License does not authorize an end user licensee to use any Sun Microsystems, Inc. name, trademark, service mark, logo or icon. The end user licensee acknowledges that Sun owns the Java trademark and all Java-related trademarks, logos and icons including the Coffee Cup and Duke ("Java Marks") and agrees to: (a) comply with the Java Trademark Guidelines at http://java.sun.com/trademarks.html; (b) not do anything harmful to or inconsistent with Sun's rights in the Java Marks; and (c) assist Sun in protecting those rights, including assigning to Sun any rights acquired by Licensee in any Java Mark.

3. Source Code. Software may contain source code that, unless expressly licensed for other purposes, is provided solely for reference purposes pursuant to the terms of your license. Source code may not be redistributed unless expressly provided for in the terms of your license.

4. Third Party Code. Additional copyright notices and license terms applicable to portions of the Software are set forth in the THIRDPARTYLICENSEREADME.txt file.

Appendix D: Service and Warranty

Service

Your Tripp Lite product is covered by the warranty described in this manual. A variety of Extended Warranty and On-Site Service Programs are also available from Tripp Lite. For more information on service, visit www.tripplite.com/support. Before returning your product for service, follow these steps:

1. Review the installation and operation procedures in this manual to insure that the service problem does not originate from a misreading of the instructions.
2. If the problem continues, do not contact or return the product to the dealer. Instead, visit www.tripplite.com/support.
3. If the problem requires service, visit www.tripplite.com/support and click the Product Returns link. From here you can request a Returned Material Authorization (RMA) number, which is required for service. This simple on-line form will ask for your unit's model and serial numbers, along with other general purchaser information. The RMA number, along with shipping instructions will be emailed to you. Any damages (direct, indirect, special or consequential) to the product incurred during shipment to Tripp Lite or an authorized Tripp Lite service center is not covered under warranty. Products shipped to Tripp Lite or an authorized Tripp Lite service center must have transportation charges prepaid. Mark the RMA number on the outside of the package. If the product is within its warranty period, enclose a copy of your sales receipt. Return the product for service using an insured carrier to the address given to you when you request the RMA.

2-Year Limited Warranty

TRIPP LITE warrants its products to be free from defects in materials and workmanship for a period of two (2) years from the date of initial purchase. TRIPP LITE's obligation under this warranty is limited to repairing or replacing (at its sole option) any such defective products. To obtain service under this warranty, you must obtain a Returned Material Authorization (RMA) number from TRIPP LITE or an authorized TRIPP LITE service center. Products must be returned to TRIPP LITE or an authorized TRIPP LITE service center with transportation charges prepaid and must be accompanied by a brief description of the problem encountered and proof of date and place of purchase. This warranty does not apply to equipment which has been damaged by accident, negligence or misapplication or has been altered or modified in any way.

EXCEPT AS PROVIDED HEREIN, TRIPP LITE MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some states do not permit limitation or exclusion of implied warranties; therefore, the aforesaid limitation(s) or exclusion(s) may not apply to the purchaser.

EXCEPT AS PROVIDED ABOVE, IN NO EVENT WILL TRIPP LITE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS PRODUCT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Specifically, TRIPP LITE is not liable for any costs, such as lost profits or revenue, loss of equipment, loss of use of equipment, loss of software, loss of data, costs of substitutes, claims by third parties, or otherwise.

Service & Warranty

Product Registration

Visit **www.tripplite.com/warranty** today to register your new Tripp Lite product. You'll be automatically entered into a drawing for a chance to win a FREE Tripp Lite product!*

* No purchase necessary. Void where prohibited. Some restrictions apply. See website for details.

WARNING!

Use of this equipment in life support applications where failure of this equipment can reasonably be expected to cause the failure of the life support equipment or to significantly affect its safety or effectiveness is not recommended. Do not use this equipment in the presence of a flammable anesthetic mixture with air, oxygen or nitrous oxide.

Regulatory Compliance Identification Numbers

For the purpose of regulatory compliance certifications and identification, your Tripp Lite product has been assigned a unique series number. The series number can be found on the product nameplate label, along with all required approval markings and information. When requesting compliance information for this product, always refer to the series number. The series number should not be confused with the marking name or model number of the product.

WEEE Compliance Information for Tripp Lite Customers and Recyclers (European Union)

Under the Waste Electrical and Electronic Equipment (WEEE) Directive and implementing regulations, when customers buy new electrical and electronic equipment from Tripp Lite they are entitled to:

- Send old equipment for recycling on a one-for-one, like-for-like basis (this varies depending on the country)
- Send the new equipment back for recycling when this ultimately becomes waste

Tripp Lite follows a policy of continuous improvement. Product specifications are subject to change without notice.

TRIPP·LITE

OVER 90 YEARS
Manufacturing
Excellence.
Since 1922.

1111 W. 35th Street, Chicago, IL 60609 USA • www.tripplite.com/support