



Lighthouse 4.5.4 User Manual

Lighthouse VM
Lighthouse Standard
Lighthouse Enterprise

revision 1.5.1
2017-09-08

Safety

Please take care to follow the safety precautions below when installing and operating the Lighthouse hardware appliance:

- 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 Opengear 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 Lighthouse hardware appliance during an electrical storm. Also it is recommended you use a surge suppressor or UPS to protect the equipment from transients.

FCC Warning Statement

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



Proper back-up systems and necessary safety devices should be utilized to protect against injury, death or property damage due to system failure. Such protection is the responsibility of the user.

This Lighthouse hardware appliance device is not approved for use as a life-support or medical system.

Any changes or modifications made to this Lighthouse hardware appliance device without the explicit approval or consent of Opengear will void Opengear of any liability or responsibility of injury or loss caused by any malfunction.

This equipment is for indoor use and all the communication wirings are limited to inside of the building.

Publishing history

Date	Revision	Update details
Feb 2013	1.0	Release manual for Lighthouse VM appliance. Software V4.0
April 2013	1.1	Added Enterprise and Standard hardware appliance. Software 4.1 (VPN, firewall)
June 2013	1.2	Updated EULA
Dec 2013	1.3	Software V4.3 (Dialpool improvements) and V4.4 (Console Gateway CLI)
Jul 2014	1.4	Software V4.5 (node-command bulk command CLI)
Feb 2015	1.5	Software V4.5.4 (Access status LED)
Sep 2017	1.5.1	Minor updates and corrections.

Copyright

© Opengear Inc. 2017. All Rights Reserved.

Information in this document is subject to change without notice and does not represent a commitment on the part of Opengear. Opengear provides this document “as is,” without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose.

Opengear may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time. This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

TABLE OF CONTENTS

THIS MANUAL	6
INITIAL DEPLOYMENT	8
2.1 Lighthouse VM Deployment	8
2.1.1 System Requirements	8
2.1.2 Distributed Files	8
2.1.3 Software deployment	8
2.1.4 Initial Login	10
2.2 Lighthouse Standard and Lighthouse Enterprise Deployment	11
2.2.1 Install the Lighthouse hardware appliance	11
CONFIGURATION	14
3.1 Welcome	14
3.2 Passwords	14
3.2.1 Enter Call Home Password	16
3.2.2 Enter License Key (VM only)	16
3.3 Configure Local Network Settings	17
3.3.1 IPv6 configuration	18
3.3.2 Dynamic DNS (DDNS) configuration	18
3.3.3 Static routes	19
3.4 Configure Managed Console Servers	20
3.4.1 Adding a Console Server	22
3.4.2 Connecting to sites on separate private or firewalled networks	23
3.5 Call Home	25
3.5.1 Setting up console server as a management candidate on CMS	25
3.5.2 Call Home to a generic central SSH server	26
3.6 Authorize Automatically Added Users	27
3.7 Authentication Configuration	28
3.7.1 Local authentication	29
3.7.2 TACACS authentication	29
3.7.3 RADIUS authentication	31
3.7.4 LDAP authentication	31
3.7.5 Group support with remote authentication	32
3.7.6 Idle timeout	33
3.7.7 Authentication testing	34
3.8 SSL Certificate	34
3.9 IPsec VPN	36
3.9.1 Enable the VPN gateway	36
3.10 OpenVPN	38
3.10.1 Enable the OpenVPN	38
3.10.2 Configure as Server or Client	39
3.10.3 Windows OpenVPN Client and Server set up	42
3.11 Firewall & Forwarding	47
3.11.1 Configuring network forwarding and IP masquerading	47
3.11.3 Port / Protocol forwarding	48
3.11.4 Firewall rules	49
3.12 Services and Service Access	51
3.13 Support Report	52
3.14 System Reset	53
3.15 Syslog	53
3.16 Dialpool – centralized dial-out	54
3.16.1 Dialpool setup	55
3.16.2 Add modems to the dialpool	55
3.16.3 Dialing Managed Console Servers	57
3.16.4 Dialpool health monitoring	59

3.17	Configuration Backup	60
3.18	Upgrade Firmware	60
3.19	Configure Date and Time	61
3.20	Key Exchange	62
3.21	Console Gateway	63
3.21.1	<i>Configuring the Console Gateway</i>	63
3.21.2	<i>Console Gateway access</i>	63
3.21.3	<i>Authentication and Authorization</i>	64
	ACCESSING MANAGED CONSOLE SERVERS & DEVICES	66
4.1	Viewing Managed Console Servers & Devices	66
4.1.1	<i>Viewing Managed Console Servers</i>	66
4.1.2	<i>Viewing Managed Devices</i>	67
4.2	Accessing Managed Console Servers & Devices	68
4.2.1	<i>Accessing Managed Console Servers</i>	68
4.2.2	<i>Accessing Managed Devices</i>	68
4.2.3	<i>Dialing Managed Console Servers</i>	70
4.3	Batch or Bulk Control of Managed Console Servers	70
4.3.1	<i>node-command Bulk CLI Command</i>	70
4.3.2	<i>node-upgrade Bulk Firmware Upgrade</i>	72
4.3.1	<i>node-user Suite Bulk User Management</i>	73
4.3.4	<i>Command Console Servers UI</i>	75
4.4	Manage Terminal	77
	MONITORING WITH NAGIOS	78
5.1	Monitor	78
5.1.1	<i>Tactical Overview</i>	78
5.1.3	<i>Services</i>	80
5.1.4	<i>Problems</i>	80
5.1.5	<i>Connecting with SDT Connector</i>	81
5.2	Reports and system	84
5.2.1	<i>Notifications</i>	84
5.3	Extended Nagios	84
5.3.1	<i>Adding custom checks + scripting/config set up</i>	85
5.3.2	<i>Introducing NagVis</i>	85
5.3.3	<i>Notifications</i>	86
5.3.4	<i>Notification Elevation</i>	87
	ACCESSING WITH SDT CONNECTOR	92
6.1	Configuring for SSH Tunneling to Hosts	93
6.2	SDT Connector client installation and configuration	93
6.3	SDT Connector to Management Console	102
6.4	SDT Connector – Telnet or SSH connect to serially attached devices	103
6.5	Using SDT Connector for out-of-band connection to the gateway	104
6.6	Importing (and exporting) preferences	105
6.7	SDT Connector Public Key Authentication	106
6.8	Setting up SDT for Remote Desktop access	106
6.9	SDT SSH Tunnel for VNC	111
6.10	Using SDT to IP connect to hosts that are serially attached to the gateway	114
6.11	SSH Tunneling using other SSH clients (e.g. PuTTY)	118
	APPENDIX A: Linux Commands & Source Code	122
	APPENDIX B: TERMINOLOGY	128
	APPENDIX C: End User License Agreement (EULA)	132
	Lighthouse hardware appliance EULA	132
	Lighthouse VM software appliance EULA	134

THIS MANUAL

This User Manual describes Opengear's Lighthouse centralized management appliance solutions and provides instructions to best take advantage of them.

- These centralized management appliances include the Lighthouse VM software appliance and the Lighthouse Standard and Lighthouse Enterprise hardware appliances. These are referred to generically in this manual as *Lighthouse* appliances
- The *Lighthouse Centralized Management System* appliances all run the same centralized management software (referred to in this manual as *CMS*). *CMS* enables network engineers and system administrators to centrally manage Opengear appliances and attached IT networking gear.
- Opengear appliances include the ACM5000, ACM5500, IM4200, CM41000 and SD4000 product lines, and they are referred to generically in this manual as *console servers*, or as *Managed Console Servers* when they are being managed by *CMS*

Who should read this guide?

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

Manual Organization

This manual contains the following chapters:

- | | |
|------------------|---|
| 1. Introduction | |
| 2. Installation | <i>Lighthouse</i> appliance and <i>CMS</i> centralized management software installation |
| 3. Configuration | Initial <i>CMS</i> configuration and connection to the <i>Managed Console Servers</i> |
| 4. Operation | Details the status displays and reports and connecting with hosts |
| 5. Nagios | Customization of the Nagios monitoring |
| 6. SDT Connector | Extended configuration options for the Java application |

The latest update of this manual can be found online at www.opengear.com/download.html.

This documentation describes using your browser to configure and operate the *Lighthouse* appliance and monitor all the connected hosts. However *Lighthouse* appliances all run a Linux operating system so experienced Linux/Nagios users may prefer to operate at the command line.

Interface icons

Icons are used in the Management Console for navigation to pages, system status, backup and restore etc.



The **logout** icon is on the top of every page. Clicking the icon logs you out and ends the current session.



Clicking the **backup** icon initiates a configuration backup



The **commit config** icon enables you to commit queued configuration changes



Click the **modify** icon to change an associated item.



You can click the **delete** icon associated with the item you want to delete.



The **cancel** icon cancels the associated item.

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 heed of these warnings

- Text presented with an arrow head 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.

Where to find additional information

The following table contains related documentation and additional sources for information.

Document	Description
Quick Start Guide	Leads you through your initial Opendgear configuration
Knowledgebase	<i>opengear.zendesk.com</i> contains a knowledgebase with technical how-to articles and tech tips

INITIAL DEPLOYMENT

OpenGear's CMS software runs on Lighthouse VM virtual software appliance platforms, and on Lighthouse Standard and Lighthouse Enterprise physical hardware platforms.

This chapter describes the initial deployment and configuration of these Lighthouse appliances.

2.1 Lighthouse VM Deployment

Lighthouse VM can be run as a guest virtual appliance under:

- Linux Kernel-based Virtual Machine (Linux KVM) or
- VMware ESX, VMware ESXi or VMware Server

The host may be a physical machine that you administer, or a managed server or a cloud hosting service from a hosting provider.

2.1.1 System Requirements

At a minimum, the Lighthouse VM requires the following reserved resources:

- 500MHz CPU core
- 256MB RAM
- 4GB disk space

The appropriate level of reserved virtual server resources will depend on the number of OpenGear appliances – and connected managed devices - being managed by the Lighthouse VM. For installations supporting 1000 or more appliances the recommended resource would be:

- 2 GHz CPU core
- 16GB RAM
- 600GB disk space

In addition, the following virtual devices are required:

- Disk device SATA (VMware) or IDE (Linux KVM)
- E1000 compatible Ethernet NIC, bridged

2.1.2 Distributed Files

The Lighthouse VM full image is released as a compressed file (*.gz) and can be downloaded from:

<http://www.opengear.com/firmware/>

Which full disk image you deploy depends on your virtualization solution:

- For Linux KVM, use: *vcms-x.y.z-kvm.hdd.gz*
- For VMware ESX/ESXi, use: *vcms-x.y.z-vmware-ovf.tar.gz*
- For VMware server, use: *vcms-x.y.z-vmware.tar.gz*

Uncompress the full image using *gunzip*, *Winzip* or similar before deployment.

Note The Lighthouse VM upgrade files (*.bin) are used for upgrades after the initial deployment. They are available from *<http://www.opengear.com/firmware/>*. Which upgrade file you use also depends on your virtualization solution. For Linux KVM, use *vcms-x.y.z-kvm.bin*. For VMware, use: *vcms-x.y.z-vmware.bin*

2.1.3 Software deployment

Follow the instructions provided by your virtualization management suite to deploy the *ovf*, *vmx* or *hdd* file as appropriate.

Examples are given below for VMware ESXi 4, VMware Workstation 7, and ElasticHosts cloud hosting provider. Further instructions on deployment can be found on the [Knowledge Base](#):

- [Deploying Lighthouse VM in a hosted Linux KVM environment](#)
- [Deploying Lighthouse VM as an ESXi virtual appliance using vSphere](#)
- [Deploying Lighthouse VM on VMware Workstation](#)

Example deployment: VMware ESXi 4

To complete this deployment you must have VMware ESXi 4 installed and running on a bare metal machine, and the VMware vSphere Client installed on a PC running Microsoft Windows. Before proceeding, download and extract the full disk image for VMware ESXi, as described in the "Distributed Files" section.

1. Launch the vSphere Client and log into the ESXi with a user who has administrator privileges.
2. In the vSphere Client, select File: Deploy OVF Template. The Deploy OVF Template wizard is displayed.
3. Specify the source location and click Next.
4. Select Deploy from File and Browse the file system for location where you extracted the contents of: vcms-x.y.z-vmware-ovf.tar.gz
Select the OVF template file, e.g.: CMS61xx-vcms-vmware.ovf
5. Check the OVF Template Details page and click Next.
6. If required, edit the OVF Template name.
7. Review the Ready to Complete details. To re-edit Source, OVF Template Details and Name and Location, click on the respective link on the left hand side of the window. Click Finish when complete.
8. The OVF Template is now displayed in the left-hand vSphere Client Status panel under the relevant host.
9. To start the virtual machine, select the Virtual Machine tab from the right-hand panel. Select the Virtual Machine by name, and click the Play button from the top menu.
10. Deployment is now complete. You can monitor the Lighthouse VM boot progress using the vSphere Client console, or proceed to "Configurion" to begin configuration.

Example deployment: VMware Workstation 7

To complete this deployment you must have VMware Workstation 7 installed and running on a PC running Microsoft Windows. Before proceeding, download and extract the full disk image for VMware Workstation, as described in the "Distributed Files" section.

1. Launch VMware Workstation.
2. Click the Launch Existing VM or Team icon in the right-hand side of the window. Browse the file system for the location where you extracted the contents of:
3. Select Deploy from File and Browse the file system for location where you extracted the contents of: vcms-x.y.z-vmware.tar.gz
Select and Open the VMX file, e.g.: CMS61xx-vcms-vmware.vmx
The Opendgear Lighthouse VM tab is displayed.
4. Click the "Power on this virtual machine" link located in the Commands box.
5. Deployment is now complete. You can monitor the Lighthouse VM boot progress using the VMware Workstation console, or proceed to "Configuring VCMS" to begin configuration.

Example Cloud Deployment: ElasticHosts

(These instructions are current as of 19 August 2010)

1. Browse to <http://www.elastichosts.com> and create an account at your preferred peer location.

2. You may wish to use the 5 day free hosting trial, otherwise add a subscription that meets the reserved resource requirements outlined under System Requirements in this document.

Ensure you set 'Committed data transfer' to 10 GB or higher and/or have pre-pay balance to cover monthly data transfer. Data usage by Lighthouse VM will vary with usage patterns, but will generally not be heavy.

We recommend you purchase a static IP address, otherwise you must also configure Lighthouse VM to use a dynamic DNS service.
3. Upload vcms-x.y.z-kvm.hdd as a drive using any of the methods described in:

`http://www.elastichosts.com/cloud-hosting/faq#uploadQ`

If you are deploying from a Linux or POSIX compliant system, we recommend using the drive upload tool script:

`http://www.elastichosts.com/downloads/elastichosts-upload.sh`

Your secret API key is available on your Profile page:


```
export EHAUTH="xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
```


Your API endpoint URI is the hostname of account's peer location, preceded by "api.", e.g. for San Antonio Peer 1:

`export EHURI=https://api.sat-p.elastichosts.com/`

After setting these in your environment, run:

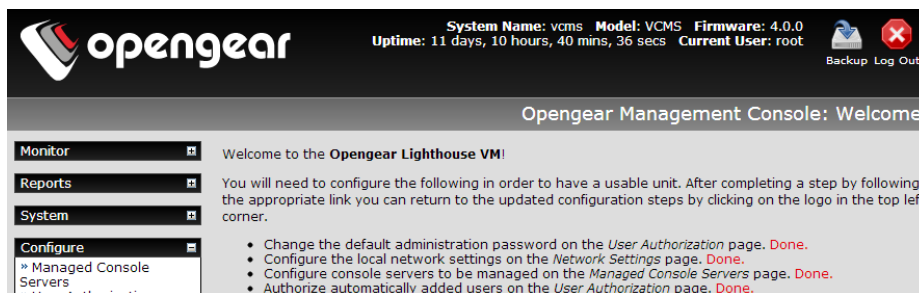
`./elastichosts-upload.sh vcms-x.y.z-kvm.hdd`
4. From the Elastic Hosts Control Panel, select Server in 'Add server or drive'. Enter a Name, e.g. "Lighthouse VM". Select the Type of 'Boot from existing drive'. Select the Drive you uploaded in the previous step, e.g. "vcms-x.y.z-kvm.hdd". Click Add.
5. Click Edit on the Server you have just added. Select the static IP address to use if available, and set the VNC password. Click Start.
6. Deployment is now complete. You can now monitor the Lighthouse VM boot progress using VNC, or proceed to "Configuring VCMS" to begin configuration

2.1.4 Initial Login

Once Lighthouse VM has been deployed and the virtual appliance has booted, configuration is performed by browsing to the IP address of the virtual NIC. The virtual NIC obtains an address using DHCP and has a static IP address of 192.168.0.1

You will be presented with the login screen on your browser

- Login as *root* with the root password
- You will then arrive at the Welcome to the **Opengear Lighthouse VM!** Screen



Note The default username / password is *root / default*. However on the initial deployment during the load process you will be prompted to enter and confirm a new root password. If you simply have upgraded to a new version you won't be prompted for a new root password, the device will just boot normally with the password it already has.

```
Starting RPC portmapper
Configuring Cron Daemon
Initialising NUT Server environment
Initialising SMS Gateway environment
Initialising Nagios Server environment
Starting CMS SSH key generation
Running product specific configuration

Welcome to your Opengear Lighthouse UM device. This is software version:
OpenGear/ Version 4.0.0 -- Fri Jan 25 16:22:27 EST 2013

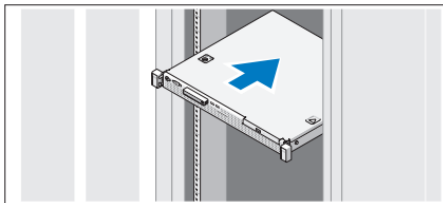
To complete initial setup, please set a new root password.
Press ENTER to continue.

Enter new root password:
Confirm given password: _
```

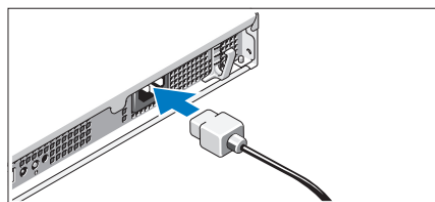
2.2 Lighthouse Standard and Lighthouse Enterprise Deployment

2.2.1 Install the Lighthouse hardware appliance

- Unpack your system and verify you have all the parts shown in the included Opengear *Quick Start Guide* - and that they all appear in good working order
- Assemble the rails and install the system in the rack following the safety instructions and the rack installation instructions provided with your system



- Proceed to connect Lighthouse Standard / Enterprise hardware appliance to the network and to power as outlined in the included Dell *Getting Started With Your System*



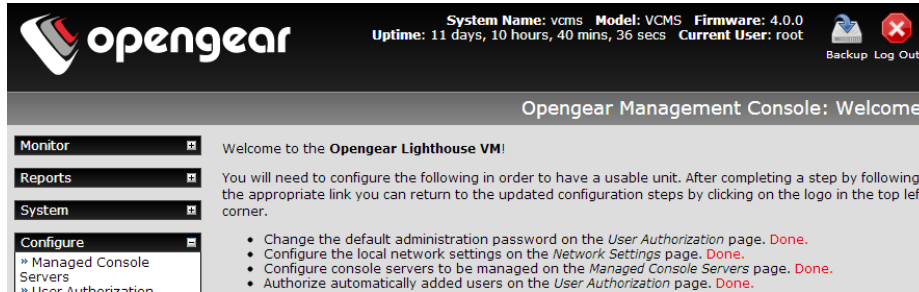
2.2.4 Initial Login

Once Lighthouse has been deployed and the hardware appliance has booted, configuration is performed by browsing to the IP address of the NIC on the Dell hardware.

The Lighthouse NIC has a default static IP address of 192.168.0.1, and your computer must have an IP address in the same network range (192.168.0.xxx). Alternately Lighthouse NIC also has its DHCP client enabled by default, so it will automatically accept any network IP address assigned by any DHCP server on your network – and will then respond at both 192.168.0.1 and its DHCP address.

You will be presented with the login screen on your browser

- Login as *root* with the root password
- You will then arrive at the Welcome to the **Opengear Lighthouse** Screen



2.2.4 Activate Dell service

The Dell hardware on which Lighthouse Standard and Enterprise is built is warranted and serviced by Dell.



It is important to register your system with Dell. This activates your support and without registration you will not receive hardware support

- Browse to <https://www.onlineregister.com/dell/>

Welcome to Dell Product Registration



Please begin registration by answering the questions below.
Asterisks * indicate required fields.

We Value Your Privacy. Dell treats your information in accordance with our [Privacy Policy](#)

* Service Tag:
 [Help](#)

* Location of purchase:



- Enter the service Tag #. This can be found on a label attached to the back on your system:



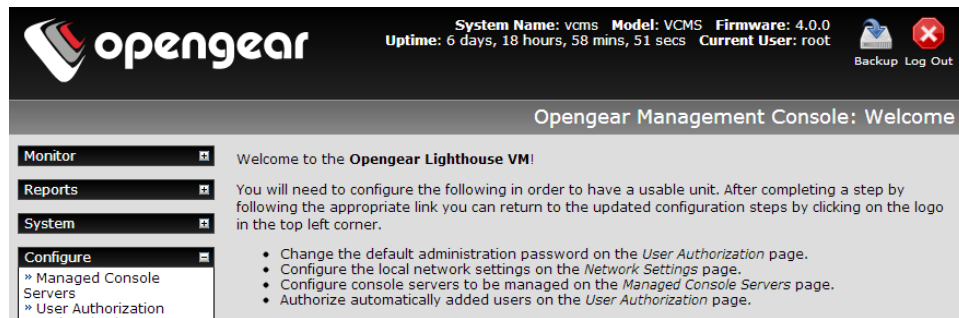
CONFIGURATION

This chapter covers the initial configuration to have a usable *Lighthouse* appliance solution.

It also discusses the other *Configure* and *Status* menu items that the *administrator* may use in managing *CMS*, such as connection to the *Managed Console Servers*, setting Time/Date and upgrading the firmware.

3.1 Welcome

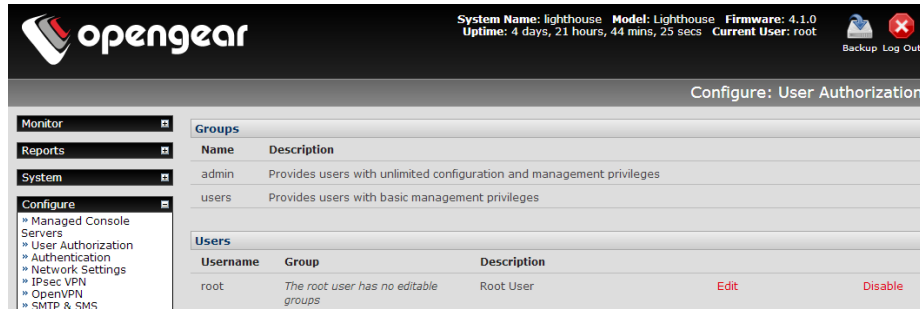
- Login as *root*. Initially only the administration user named **root** can log into *CMS*. The factory-shipped root password is *default*, However you will have changed this on initial deployment



- You will arrive at the Welcome to the **Opengear Lighthouse VM!** screen. Follow the initial configuration steps:
 - Enter new passwords (Chapter 3.2).
 - Configure the local network settings (Chapter 3.3)
 - Configure console servers to be managed (Chapter 3.4)
 - Authorize automatically added users (Chapter 3.6)
- After completing a step (by following the appropriate link) you can return to the updated configuration steps by clicking on the logo in the top left corner

3.2 Passwords

- Before changing the default administration password, turn the https service on. This ensures the changed password string is sent between the browser and the *console server* in encrypted fashion.
- Choose **System: Services**.
- Check the *Enable HTTPS Web Management* checkbox.
- Click **Apply**.
- Click the Opengear logo to return to the Welcome page.
- *Change the default administration password* takes you to **Configure: User Authorization**, where you can again reset the password for **root**

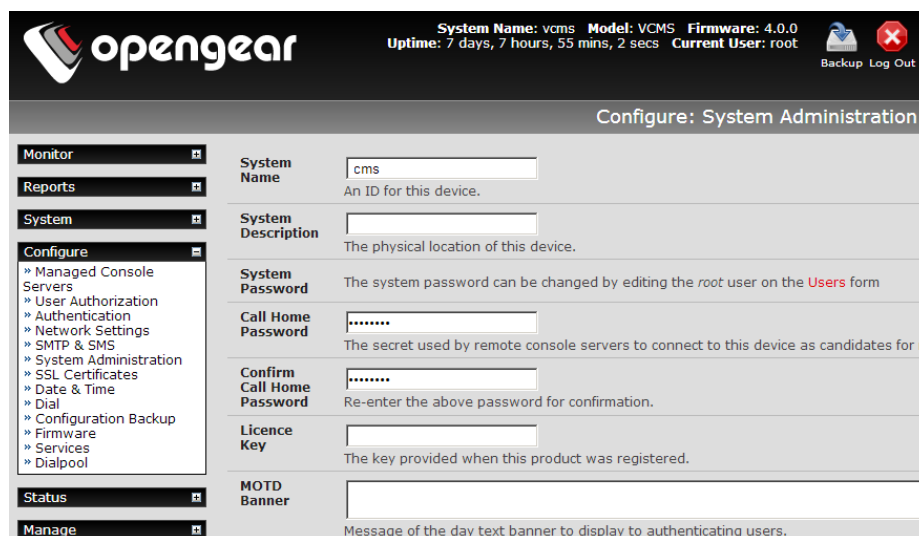


Note With Lighthouse VM you will have already been prompted to change the root password during initial deployment so this step may already be *Done*. However this is the main administrative user account, so it is important that you choose a complex password, and keep it safe.

- Enter a new **System Password** then re-enter it in **Confirm System Password**
- Click **Apply**. If you have changed the password you will be prompted to log in again. This time use the new *System Password*



- Select **Configure: System Administration** to now enter other passwords



- At this stage you may also wish to enter a **System Name** and **System Description** to give your *Lighthouse* appliance a unique ID and make it simple to identify

Note The System Name can contain from 1 to 64 alphanumeric characters. You can also use the special characters "-" "_" and "."). Similarly there are no restrictions on the characters that can be used in the System Description or the System Password. Each of these can contain up to 254 characters.

- Click **Apply**

3.2.1 Enter Call Home Password

If you wish to monitor *Managed Console Servers* that are connected via Call Home, you will need a Call Home password:

- Enter a new **Call Home Password** then re-enter it in **Confirm Call Home Password**
- Click **Apply**

This password is used for a system account used solely for accepting Call Home connections. It is safe to change this password, without affecting currently established Call Home connections.

Note If you use remote authentication without any fallback to local authentication checks the 'cms' user authentication will fail if you don't have a 'cms' user in the remote authentication. This authentication failure will cause the set-up of a new Call Home console server to fail.

3.2.2 Enter License Key (VM only)

When you ordered your Lighthouse VM license you will have been emailed a License Key. Install this key now- before proceeding with the configuration steps (which are described in the next chapter).

To install the Key:

- Copy the Key from the email that you received into the **Licence Key** field
- Click **Apply**

This step is required for Lighthouse VM only. With the Lighthouse Enterprise and Lighthouse Standard the Key will have been pre-installed

The screenshot shows the OpenGear configuration interface. At the top, the system status is displayed: System Name: vcms, Model: VCMS, Firmware: 4.0.0, Uptime: 7 days, 7 hours, 55 mins, 2 secs, and Current User: root. There are Backup and Log Out buttons in the top right corner. The main content area is titled 'Configure: System Administration' and contains several configuration fields:

- System Name:** cms (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)
- Call Home Password:** (The secret used by remote console servers to connect to this device as candidates for...)
- Confirm Call Home Password:** (Re-enter the above password for confirmation.)
- Licence Key:** (The key provided when this product was registered.)
- MOTD Banner:** (Message of the day text banner to display to authenticating users.)

Note This License Key provides you with a commercial license to use the Lighthouse VM software appliance to manage up to the designated number of appliances for the defined period. For example ordering an OGLH-VM-100-3Y license enables you to use your *Lighthouse* appliance to manage a distributed network with up to 100 Opengear appliances with support and feature upgrades for 3 years. You can then renew your License Key

annually to receive ongoing support and upgrades. If you have to contact support, they will ask you to quote the Licence Key number from this page.

3.3 Configure Local Network Settings

The next step is to enter an IP address and network settings for the *Network* port on the *CMS*, or to enable its DHCP client so that it automatically obtains an IP address from a DHCP server on the network it is to be connected to

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

The screenshot shows the OpenGear CMS interface for configuring network settings. The 'Configure: Network Settings' page is active, with the 'Network Interface' tab selected. The 'Configuration Method' is set to 'DHCP'. Below this, there are input fields for 'IP Address', 'Subnet Mask', 'Gateway', 'Primary DNS', and 'Secondary DNS'. The 'IP Address' field is currently empty, while the others contain the values 208.67.222.222. A sidebar on the left contains navigation menus for Monitor, Reports, System, Configure, Status, and Manage. The top of the page displays system information: System Name: vcms, Model: VCMS, Firmware: 4.0.0, Uptime: 8 days, 11 hours, 7 mins, 54 secs, Current User: root, and buttons for Backup and Log Out.

- If you selected **DHCP** the *CMS* will look for configuration details from a DHCP server on your management LAN. This selection automatically disables any static address

Note In its factory default state (with no Configuration Method selected) the *CMS* 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 *CMS* will then respond to both its Static address (192.168.0.1) and its newly assigned DHCP address

- By default the *CMS* Network 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 *CMS* 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

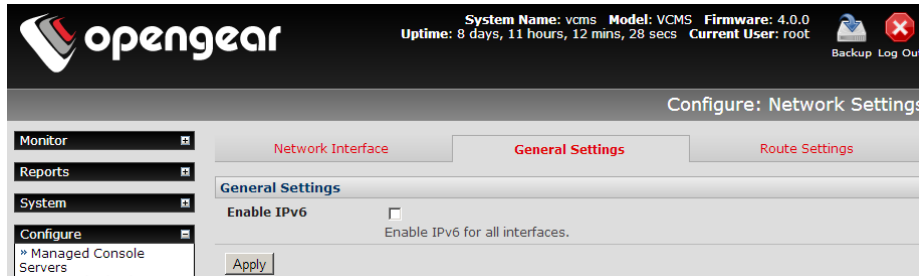
- Click **Apply**
- You will need to reconnect the browser on the PC/workstation that is connected to the *CMS* by entering **https://new IP address**

Note If you selected the DHCP configuration method, and plan to use Call Home **it is strongly recommended** that you use a dynamic DNS service. So at this point, you may also configure dynamic DNS. For detailed setup instructions, see the sections entitled Call Home and Dynamic DNS later in this document.

3.3.1 IPv6 configuration

The CMS Network interface can also be configured for IPv6 operation:

- On the **Configure: Network Settings** menu select **General Settings** page and check **Enable IPv6**



3.3.2 Dynamic DNS (DDNS) configuration

With Dynamic DNS (DDNS), an appliance whose IP address is dynamically assigned (and that may change from time to time) can be located using a fixed host or domain name.

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

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. By default DDNS is disabled. To enable:

- On the **Configure: Network Settings** menu select the **Network Interface** page then select the DDNS service provider from the drop down **Dynamic DNS** list

Dynamic DNS	
Dynamic DNS	<input type="text" value="None - DDNS disabled"/> Update a DNS server when IP address is changed.
DDNS update server	<input type="text"/> The DDNS server to push updates to. The format is server address:port <i>This is used by gnuddp only</i>
DDNS Hostname	<input type="text"/> The Fully Qualified DNS hostname assigned to this interface.
DDNS Username	<input type="text"/> The username for the account to manage this interface.
DDNS Password	<input type="password"/> The password for the account to manage this interface.
Confirm DDNS Password	<input type="password"/> Re-enter the password for confirmation.
Maximum interval between updates	<input type="text"/> Maximum interval between updates in days. DDNS update will be sent even if the address has not changed. <i>Defaults to 25.</i>
Minimum interval between checks	<input type="text"/> Minimum interval between checks for changed addresses, in seconds. Updates will still only be sent if the address has changed. <i>Defaults to 1800.</i>
Maximum attempts per update	<input type="text"/> Number of times to attempt an update before giving up. <i>Defaults to 3.</i>
<input type="button" value="Apply"/>	

- 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)

3.3.3 Static routes

Route Settings enables you to set up *static routes* which provide a very quick way to route data from one subnet to different subnet. So you can hard code a path that specifies to the *CMS/router* to get to a certain subnet by using a certain path. This may be useful for remotely accessing various subnets at a remote site when being accessed using the cellular OOB connection.

The screenshot shows the OpenGear configuration interface. At the top, it displays system information: System Name: vcms, Model: VCMS, Firmware: 4.0.0, Uptime: 8 days, 11 hours, 23 mins, 6 secs, and Current User: root. There are also buttons for Backup and Log Out. The main navigation menu on the left includes Monitor, Reports, System, and Configure. The Configure menu is expanded, showing options like Managed Console Servers, User Authorization, Authentication, Network Settings, SMTP & SMS, System Administration, SSL Certificates, Date & Time, Dial, Configuration Backup, Firmware, Services, and Dialpool. The main content area is titled 'Configure: Network Settings' and has three tabs: Network Interface, General Settings, and Route Settings. The Route Settings tab is active, showing a form with the following fields: Route Name (New Route), Destination Network/Host, Destination netmask (24), Route Gateway, and Metric (0). An Apply button is at the bottom of the form.

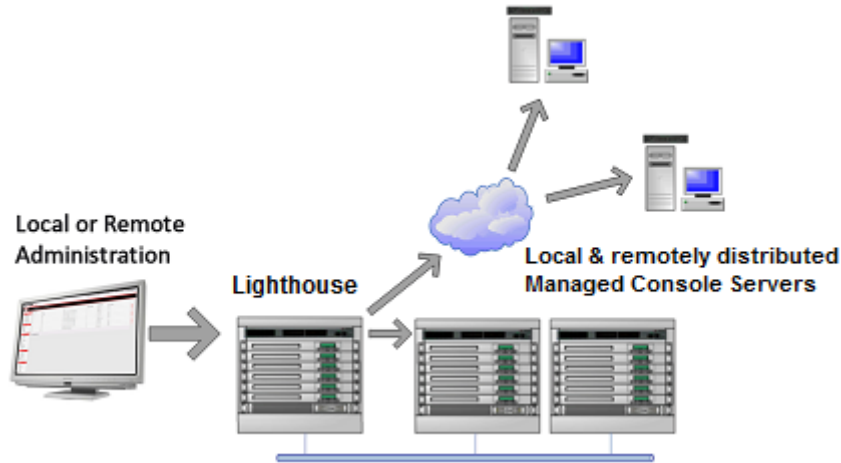
To add to the static route to the route table of the system:

- 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. Any number between 0 and 32. A subnet mask of 32 identifies a host route.
- Enter **Route Gateway** with 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. Any number equal to or greater than 0
- Click **Apply**

3.4 Configure Managed Console Servers

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

To manage Local Console Servers, or console servers that are reachable from the *CMS*, the SSH connections are initiated by *CMS*. To manage Remote Console Servers, or console servers that are firewalled, not routable, or otherwise unreachable from the *CMS*, the SSH connections are initiated by the *Managed Console Server* via an initial Call Home connection. This ensures secure, authenticated communications and enables *Managed Console Server* units to be distributed locally on a LAN, or remotely around the world.



➤ **Select Configure: Managed Console Servers**

The *Managed Console Servers* list displays all the console servers which are currently being monitored by the CMS:

- The *Managed Device Last Retrieved* field shows when each console server's configuration information (such as user and *Managed Device* details, alert settings etc) was last updated in the CMS. To update this information check the *Managed Console Server(s)* to be updated and click **Retrieve Hosts**
- The *IP Address/DNS Name* shows how the CMS is accessing this *Managed Console Server*.
 - For a Local Console Server, it shows the network address and SSH server port that CMS is connected to
 - For a Remote Console Server, it shows the local redirected port, and the remote IP address from which the connection has originated. The local redirected port matches the Listening Port as displayed in the Call Home connection on the Remote Console Server

opengear System Name: vcms Model: VCMS Firmware: 4.0.0
 Uptime: 8 days, 11 hours, 56 mins, 57 secs Current User: root Backup Log Out

Configure: Managed Console Servers

Name	IP Address/DNS Name	Description	Managed Devices Last Retrieved
<input type="checkbox"/> sd4001	Port 61637 (localhost:61637 → 76.118.187.59)	sd4001	Fri Jan 25 14:11:29 2013 <input type="button" value="Edit"/>
<input type="checkbox"/> ACM5004gatt	Port 50405 (localhost:50405 → not connected)	ATT	Tue Nov 13 20:01:18 2012 <input type="button" value="Edit"/>

Select/unselect all nodes

Detected Console Servers

Local Console Servers No local console servers detected.

Remote Console Servers No remote console servers detected.

New Console Server
 Manually enter the details of a console server to manage.

3.4.1 Adding a Console Server

The *Detected Console Servers* list displays all the *console servers* which are currently not being monitored by the *CMS*:

- The *Local Console Servers* drop down list shows all the *console servers* which are on the same subnet as the *CMS*, and are not currently being monitored. Click **Refresh** to update
- The *Remote Console Servers* drop down list shows all the *console servers* that have established a Call Home connection (so are candidates) but are not currently being monitored. Click **Refresh** to update

Note When adding a (Detected) Remote Console Server, the IP Address will appear as localhost. This is the loopback listening port created by the Call Home connection

- To add a *console server* to the *Managed Console Servers* list, either select it from the Local or Remote Console Servers drop down list, and click **Add**

Note Alternately you can manually add a *console server* to the *Managed Console Server* list by entering its details in the **New Console Server** section. You may wish to do this if the *console server* is at a remote address, but is reachable from the *CMS* – and you do not wish to use Call Home. Simply specify the SSH server address and port of the *console server* and click **Add**

- Enter the **IP Address /DNS Name** and **SSH Port** if these fields have not been auto-completed
- Enter a **Description** and unique **Name** for the *Managed Console Server* you are adding (e.g. “Boston”)

The screenshot shows the OpenGear CMS interface. At the top, it displays system information: System Name: vcms, Model: VCMS, Firmware: 4.0.0, Uptime: 8 days, 12 hours, 13 mins, 48 secs, Current User: root. There are also icons for Backup and Log Out. The main navigation sidebar includes Monitor, Reports, System, Configure (with sub-items like Managed Console Servers, User Authorization, Authentication, Network Settings, SMTP & SMS, System Administration, SSL Certificates, Date & Time, Dial, Configuration Backup, Firmware, Services, Dialpool), Status, and Manage. The main content area is titled 'Configure: Managed Console Servers' and contains a form with the following fields:

- Name**: Short name to identify the managed console server.
- Description**: A brief description of the managed console server.
- IP Address/DNS Name**: The managed console server's IP address or DNS name.
- SSH Port**: The managed console server's SSH server port.
- Remote Root Password**: The root password set on the managed console server. This password will not be stored, but used to propagate SSH keys and then forgotten.
- SSH Key Fingerprint**: Unknown, enter IP Address/DNS Name and SSH Port then click [here](#) to retrieve fingerprint.

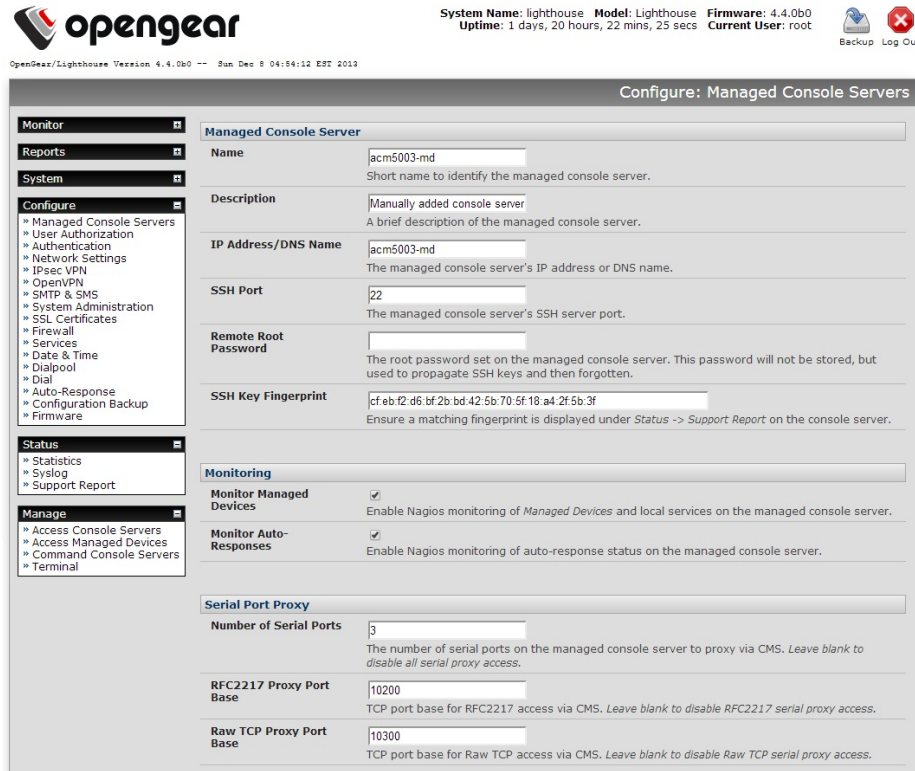
Below the form is a **Monitoring** section with a checkbox for **Monitor Managed Devices**, which is checked. The description for this checkbox is: 'Enable Nagios monitoring of Managed Devices and local services on the managed console server.'

- Enter the **Remote Root Password** (i.e. System Password that has been set on this *Managed Console Server*)

Note This password is used by the *CMS* to propagate auto generated SSH keys and then forgotten. This password will not be stored

- Check **Monitor Managed Devices** to enable Nagios monitoring of Managed Devices and local services on the managed console server
- Check **Monitor Auto-Responses** to enable Nagios monitoring of auto-response status on the managed console server

- The **Serial Port Proxy** sets the number of ports on the Managed Console Server the CMS has proxy access to.
- Add the **RFC2217 Proxy Port Base** when you want CMS to act as a single point for virtual com port access (eg as a *Console Gateway* as described in Section 3.21). Setting the number of ports also determines how many Ajax Webterms are accessible from the *Access Console Server* page



System Name: lighthouse Model: Lighthouse Firmware: 4.4.0b0
Uptime: 1 days, 20 hours, 22 mins, 25 secs Current User: root

OpenGear/Lighthouse Version 4.4.0b0 -- Sun Dec 8 04:54:12 EST 2013

Configure: Managed Console Servers

Managed Console Server

Name: acm5003-md
Short name to identify the managed console server.

Description: Manually added console server
A brief description of the managed console server.

IP Address/DNS Name: acm5003-md
The managed console server's IP address or DNS name.

SSH Port: 22
The managed console server's SSH server port.

Remote Root Password: [Redacted]
The root password set on the managed console server. This password will not be stored, but used to propagate SSH keys and then forgotten.

SSH Key Fingerprint: cf:eb:f2:d6:bf:2b:bd:42:5b:70:5f:18:a4:2f:5b:3f
Ensure a matching fingerprint is displayed under Status -> Support Report on the console server.

Monitoring

Monitor Managed Devices:
Enable Nagios monitoring of *Managed Devices* and local services on the managed console server.

Monitor Auto-Responses:
Enable Nagios monitoring of auto-response status on the managed console server.

Serial Port Proxy

Number of Serial Ports: 3
The number of serial ports on the managed console server to proxy via CMS. Leave blank to disable all serial proxy access.

RFC2217 Proxy Port Base: 10200
TCP port base for RFC2217 access via CMS. Leave blank to disable RFC2217 serial proxy access.

Raw TCP Proxy Port Base: 10300
TCP port base for Raw TCP access via CMS. Leave blank to disable Raw TCP serial proxy access.

- For details on **Remote Dialin Setup** refer subsequent **Dialpool** section
- Click **Apply**.

The CMS will now set up secure SSH connections to and from the *Managed Console Server*. It will be included in the *Managed Console Servers* list (which displays all the console servers which are currently being monitored by the CMS). And the CMS will retrieve its *Managed Devices*, user account details and configured alerts.

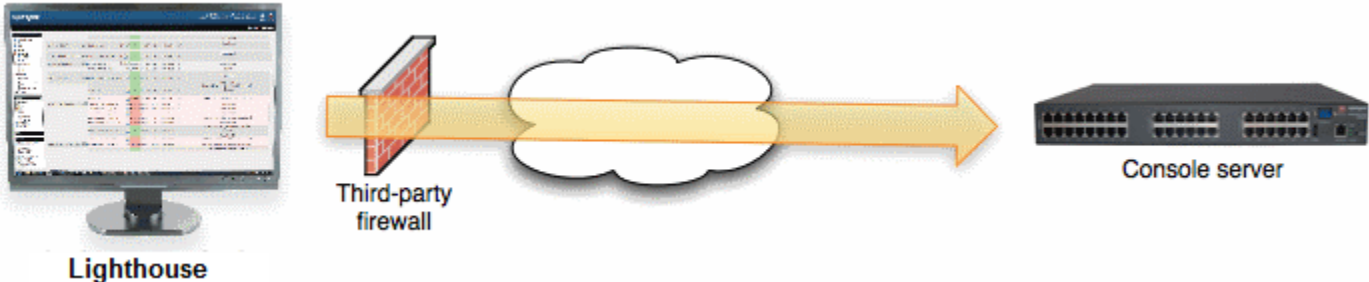
3.4.2 Connecting to sites on separate private or firewalled networks

Often, the remote *console servers* - or the *Lighthouse* appliance itself - will be on private firewalled networks. So they are unable to directly connect to each other.

Whatever the topology, as long as either CMS can SSH to the console server or the *console server* can SSH to CMS, then the CMS can manage the console server.

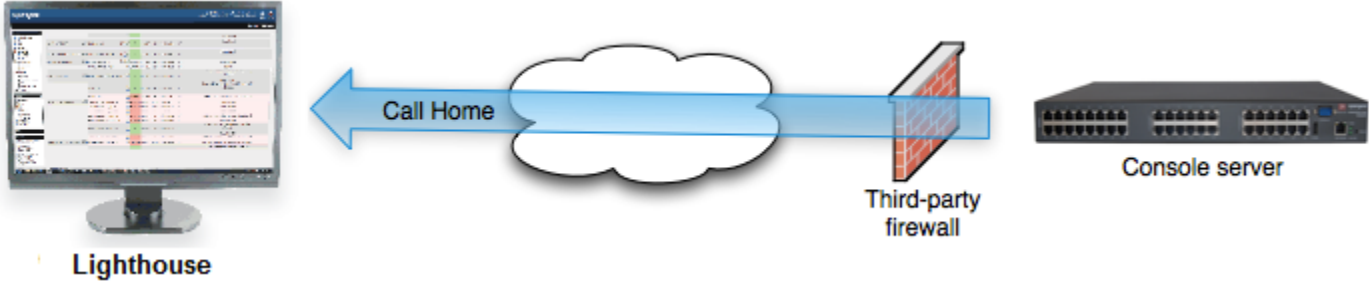
There are three common scenarios:

- I. **The console server has a public address and the CMS has a private or firewalled address.**



In this case, ensure the third-party firewall allows outbound connections the distributed console server's SSH port (outbound destination TCP port 22). This is the default behavior of most firewalls. The distributed console server will not be detected by the CMS, but can be added manually at the CMS using *Configure -> Managed Console Servers -> New Console Server -> Add* as described above.

II. The console server has a private or firewalled address and the CMS has a public address.



This is a common for console servers using cellular connections. On the console server, use Serial & Network -> Call Home to connect the console server to the CMS public address. The distributed console server will then be detected by the CMS and can be added using *Configure -> Managed Console Servers -> Remote Console Servers* as described in the next section

III Both the console server and CMS have a private or firewalled address.

There are two options in this scenario:

(a) Make CMS accessible by the console servers

This is usually the preferable option if there are multiple console servers with private or firewalled addresses - common with console servers using cellular connections connecting to a CMS on a central private operations network.



Configure the third-party firewall to port forward (PAT) from its public address to the CMS's private address, targeting TCP port 22. The public forwarded port may be any port, e.g. 2222.

Configure the CMS with the external IP or DNS address of the third-party firewall. Connect to the CMS command line using SSH and run:

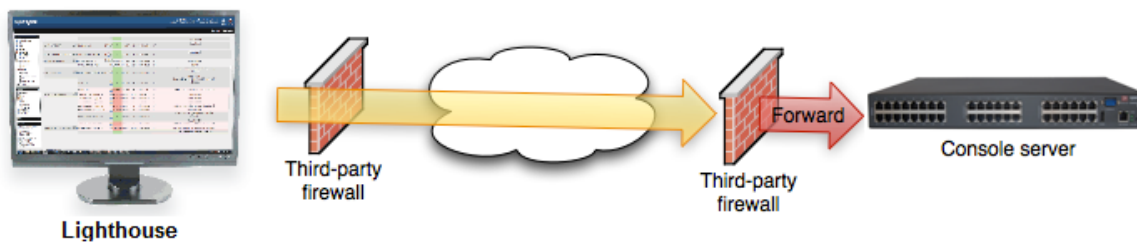
```
config -s config.cms.address=4.3.2.1 config -s config.cms.sshport=2222
```

... where 4.3.2.1 is public address of the third-party firewall, and 2222 is the public forwarded port.

Once this is done, the managed console server can Call Home to the CMS using the forwarded port as per scenario 2 above.

(b) Make the console server accessible by CMS

Configure the third-party firewall to port forward (PAT) from its public address to the console server's private address, targeting TCP port 22.



The public forwarded port may be any port, e.g. 1022, 2022 - this allows for multiple console servers to be managed behind a single firewall. Once this is done, add the managed console server to CMS as described in the earlier section.

3.5 Call Home

To manage a console server, the *CMS* must be able to connect to it using SSH. Sometimes this is not possible, e.g. if a console server is behind a third party firewall, or has a private, non-routable IP address. This is often the case when the console server is connected via a Cellular Modem connection.

In this situation, a Call Home connection can be initiated from the console server to the *CMS*. This creates an SSH listening port on the *CMS*, that is redirected back across the Call Home connection to the console server. This allows the *CMS* to connect to the console server using SSH, and thereby manage it.

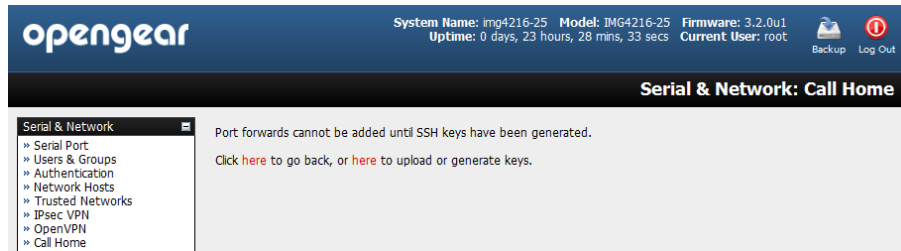
Any *console server* with Firmware V3.2 or later, has *Call Home* support.

Note To Call Home, the console server must be able to connect to the *CMS* using SSH. It is also important that the *CMS* has a static IP address. If this is not possible, you must configure the *CMS* to use a dynamic DNS service (refer Dynamic DNS section later in this manual).

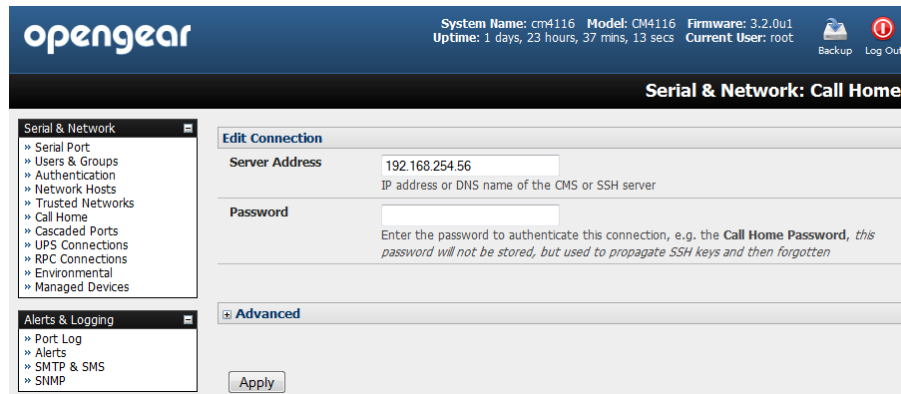
3.5.1 Setting up console server as a management candidate on CMS

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

- Browse to the *console server's* management console and select **Call Home** on the **Serial & Network** menu



- If you have not already generated or uploaded an SSH key pair for this *console server*, you will need to do so before proceeding. Details on this procedure are outlined in the Opengear User Manual in the section entitled *Automatically generate and upload SSH keys*
- Click **Add**



- Enter the IP address or DNS name (e.g. the dynamic DNS address) of the *CMS*
- Enter the Password that you configured on the *CMS* as the **Call Home Password**
- Click **Apply**



These steps initiate the Call Home connection from the *console server* to the *CMS*. An SSH listening port is created on the *CMS*, and the *console server* is set up as a candidate to be accepted as a *Managed Console Server*.

Once the candidate has been accepted on the *CMS* (as outlined in the previous section), an SSH tunnel to the *console server* is then redirected back across the Call Home connection. The *console server* has now become a *Managed Console Server* and the *CMS* can connect to and monitor it through this tunnel.

3.5.2 Call Home to a generic central SSH server

If you are connecting to a generic SSH server (not a *CMS*), you may configure *Advanced* settings:

- Enter the **SSH Server Port** and SSH User to authenticate as

- Enter the details for the SSH port forward(s) to create

System Name: cm4116 Model: CM4116 Firmware: 3.2.0u1
Uptime: 1 days, 23 hours, 37 mins, 13 secs Current User: root Backup Log Out

Serial & Network: Call Home

Edit Connection

Server Address: 192.168.254.56
IP address or DNS name of the CMS or SSH server

Password:
Enter the password to authenticate this connection, e.g. the Call Home Password, this password will not be stored, but used to propagate SSH keys and then forgotten

Advanced

SSH Server Port: 22
The SSH server port

SSH User: cms
User to authenticate as

Listening Port

Listening Server	Listening Port	Target Server	Target Port	
<input checked="" type="radio"/> Remote	57452	127.0.0.1	22	Remove
<input type="radio"/> Local				

Add

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

- Specify a Listening Port to forward from, leave this field blank to allocate an unused port
- Enter the Target Server and Target Port that will be the recipient of forwarded connections

3.6 Authorize Automatically Added Users

CMS retrieves and aggregates user accounts that are locally configured on *Managed Console Servers*. This way, a user with accounts across multiple *Managed Console Servers* has a single pane of glass from which they can monitor and access all the *Managed Console Servers* and subordinate *Managed Devices* the user has permissions to access.

Once a user account has been retrieved for the first time, it must be explicitly authorized on the CMS before that user can log in to the CMS.

- Select **Configure: User Authorization**. This will display a list of all the users which have been set up on all the *Managed Console Servers* currently being monitored by the CMS

System Name: vcms Model: VCMS Firmware: 4.0.0
Uptime: 8 days, 13 hours, 15 mins, 8 secs Current User: root Backup Log Out

Configure: User Authorization

Groups

Name	Description
admin	Provides users with unlimited configuration and management privileges
users	Provides users with basic management privileges

Users

Username	Group	Description	Edit	Disable
root		The root user has no editable groups Root User	Edit	Disable
jared	users	Retrieved User	Edit	Disable
Joe	users	Unauthorized user, click Edit to set password	Edit	Disable

- For any user, select **Edit** and enter a new password that will be used by that user when accessing *CMS*
- At this stage, you can also modify the *Group* membership and *Description* associated with that particular user. Users in the **user** group can access the *Current Status* menus, the *Reports* menus and the *System* menu (basically all the monitoring screens) whereas users in the **admin** group have this access plus the ability to reconfigure the *CMS* using the *Configure* menu
- Enter the user's **Email Address** to be used for sending notifications
- An SMS alert can also be sent via an SMTP (email) gateway. You will need to specify the **SMTP SMS Email Address** in the format specified by the gateway provider e.g. for T-Mobile it is *phonenumber@tmomail.net*

Note Group membership on the *CMS* is distinct from group members on *Managed Console Servers*. Groups set on *CMS*, control access to the *CMS* only, and are not retrieved from or propagated to *Managed Console Servers*.

- Click **Apply**

The screenshot shows the OpenGear CMS configuration interface. At the top, the system information is displayed: System Name: vcms, Model: VCMS, Firmware: 4.0.0, Uptime: 8 days, 13 hours, 20 mins, 40 secs, Current User: root. The page title is "Configure: User Authorization". The left sidebar contains a navigation menu with "Configure" expanded to "User Authorization". The main content area is titled "Edit an Existing User" and contains several form fields: Username (Joe), Description (Retrieved User), Email Address, SMTP SMS Email Address, Groups (admin and users), Password, and Confirm. The "users" group is selected. The "Apply" button is at the bottom.

3.7 Authentication Configuration

Authentication can be performed locally, or remotely using an *LDAP*, *Radius* or *TACACS+* authentication server. The default authentication method for the *CMS* is *Local*.



Any authentication method that is configured will be used for authentication of any user who attempts to log in through HTTPS or SSH to the CMS.

The CMS can be configured to the default (**Local**) or an alternate authentication method (**TACACS**, **RADIUS** or **LDAP**) with the option of a selected order in which local and remote authentication is to be used:

Local TACACS /RADIUS/LDAP: Tries local authentication first, falling back to remote if local fails

TACACS /RADIUS/LDAP Local: Tries remote authentication first, falling back to local if remote fails

TACACS /RADIUS/LDAP 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)

3.7.1 Local authentication

- Select **Configure: Authentication** and check **Local**
- Click **Apply**

3.7.2 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 **Configure: Authentication** and check **TACAS** or **LocalTACACS** or **TACACSLocal** or **TACACSDownLocal**

TACACS+	
Authentication and Authorisation Server Address	<input type="text" value="test-linux"/> Comma separated list of remote authentication and authorization servers.
Accounting Server Address	<input type="text"/> Comma separated list of accounting remote accounting servers. If unset, authentication and authorization server addresses will be used.
Server Password	<input type="password" value="....."/> The shared secret allowing access to the authentication server
Confirm Password	<input type="password" value="....."/>
TACACS Login Method	<input type="radio"/> PAP <input type="radio"/> CHAP <input type="radio"/> Login The method used to authenticate to the server. Defaults to PAP. <i>To use DES encrypted passwords, select Login</i>
TACACS Group Membership Attribute	<input type="text"/> The TACACS attribute that is used to indicate group memberships. Defaults to: groupname#n
TACACS Service	<input type="text"/> The service to authenticate with. This determines which set of attributes are returned by the server. Defaults to <i>raccess</i>
Default Admin Privileges	<input type="checkbox"/> Enable to give all TACAS+ authenticated users <i>admin</i> privileges. Use Remote Groups must be ticked for the privileges to be granted

- 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 and confirm the **Server Password**. Then select the method to be used to authenticate to the server (defaults to **PAP**). To use DES encrypted passwords, select **Login**
- If required enter the **TACACS Group Membership Attribute** that is to be used to indicate group memberships (defaults to *groupname#n*)
- If required, specify **TACACS Service** to authenticate with. This determines which set of attributes are returned by the server (defaults to *raccess*)
- If required, check **Default Admin Privileges** to give all TACAS+ authenticated users *admin* privileges. **Use Remote Groups** must also be ticked for these privileges to be granted
- Click **Apply**. TACACS+ 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

3.7.3 RADIUS authentication

Perform the following procedure to configure the RADIUS authentication method to be used whenever the *CMS* or any of its serial ports or hosts is accessed:

- Select **Configure: Authentication** and check **RADIUS, LocalRADIUS, RADIUSLocal** or **RADIUSDownLocal**

RADIUS	
Authentication and Authorisation Server Address	<input type="text"/> Comma separated list of remote authentication and authorisation servers.
Accounting Server Address	<input type="text"/> Comma separated list of remote accounting servers. If unset, Authentication and Authorisation Server Address will be used.
Server Password	<input type="text"/> The shared secret allowing access to the authentication server.
Confirm Password	<input type="text"/> 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 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 *CMS* 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-eeed-49e4-88f6-9e304f97fetc.mspx>

http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a00800945cc.shtml

<http://www.freeradius.org/>

3.7.4 LDAP authentication

Perform the following procedure to configure the LDAP authentication method to be used whenever the *CMS* or any of its serial ports or hosts is accessed:

- Select **Configure: Authentication** and check **LDAP, LocalLDAP, LDAPLocal** or **LDAPDownLocal**

LDAP	
Server Address	<input type="text"/> Comma separated list of remote servers.
Server Password	<input type="text"/> The shared secret allowing access to the authentication server.
Confirm Password	<input type="text"/> Re-enter the above password for confirmation.
LDAP Base DN	<input type="text"/> The distinguished name of the search base. For example: dc=my-company,dc=com
LDAP Bind DN	<input type="text"/> The distinguished name to bind to the server with. The default is to bind anonymously.
LDAP Username Attribute	<input type="text"/> The LDAP attribute corresponding to the login name. On Active Directory servers, the attribute is sAMAccountName
LDAP Group Membership Attribute	<input type="text"/> The LDAP attribute that is used to indicate group memberships. On Active Directory servers, the attribute is memberOf
LDAP Console Server Group DN	<input type="text"/> 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	<input type="text"/> The distinguished name of a group existing on the server whose members will be given admin access

- 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**

Note To interact with LDAP requires that the user account exists on our *CMS* to work with the remote server i.e. you can't just create the user on your LDAP server and not tell the *CMS* about it. You need to add the user account.

- Click **Apply**. LDAP remote authentication will now be used for all user access to *CMS* and serially or network attached devices

LDAP The Lightweight Directory Access Protocol (LDAP) is based on the X.500 standard, but significantly simpler and more readily adapted to meet custom needs. The core LDAP specifications are all defined in RFCs. LDAP is a protocol used to access information stored in an LDAP server. 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/>

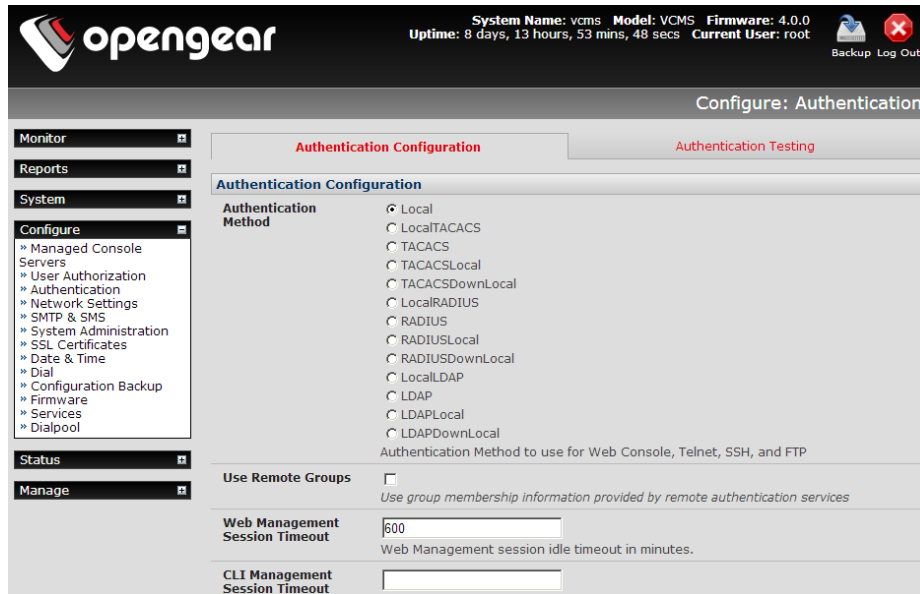
3.7.5 Group support with remote authentication

CMS allows remote authentication via RADIUS, LDAP and TACACS+. RADIUS and LDAP can provide additional restrictions on user access based on group information or membership. For example, with remote group support, users can belong to a local group that has been 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 **Configure: Authentication**
- Select the relevant **Authentication Method**
- Check the **Use Remote Groups** button



- Refer to your *console server* User Guide for remote group configuration details

3.7.6 Idle timeout

You can specify amount of time in minutes the CMS waits before it terminates an idle ssh or web connection.

- Select **Configure: 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

3.7.7 Authentication testing

The Authentication Testing enables the connection to the remote authentication server to be tested.

3.8 SSL Certificate

The CMS uses the Secure Socket Layer (SSL) protocol for encrypted network traffic between itself and a connected user. During the connection establishment the CMS has to expose its identity to the user's browser using a cryptographic certificate. The default certificate that comes with the CMS device upon delivery is for testing purposes 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 the Internet

- Activate your preferred browser and enter `https:// IP address`. 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.

However, it is recommended you generate and install a new base64 X.509 certificate that is unique for a particular *CMS*.

The screenshot shows the OpenGear web interface for configuring SSL certificates. The top navigation bar includes the OpenGear logo, system information (System Name: vcms, Model: VCMS, Firmware: 4.0.0, Uptime: 8 days, 14 hours, 37 mins, 10 secs, Current User: root), and Backup/Log Out buttons. The main content area is titled 'Configure: SSL Certificates' and contains a left-hand navigation menu with categories: Monitor, Reports, System, Configure (with sub-items like Managed Console Servers, User Authorization, Authentication, Network Settings, SMTP & SMS, System Administration, SSL Certificates, Date & Time, Dial, Configuration Backup, Firmware, Services, Dialpool), Status, and Manage. The main form fields are: Common name (text input), Organizational unit (text input), Organization (text input), Locality/City (text input), State/Province (text input), Country (dropdown menu set to AD), Email (text input), Challenge Password (text input), Confirm Password (text input), and Key Length (bits) (dropdown menu set to 512).

To do this the *CMS* 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 *CMS*:

- Select **System: SSL Certificate** and fill out the fields as explained below:

Common name This is the network name of the *CMS* once it is installed on the network (usually the fully qualified domain name). It is identical to the name that is used to access the *CMS* 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 *CMS* is accessed using HTTPS

Organizational Unit This field is used for specifying to which department within an organization the *CMS* belongs

Organization The name of the organization to which the *CMS* 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. This country code has to be entered in CAPITAL LETTERS

Email The email address of a contact person that is responsible for the *CMS* 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 *CMS* 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 *CMS* using the **Upload** button as shown below

After completing these steps the *CMS* will have its own certificate that is used for identifying the *CMS* to its users.

3.9 IPsec VPN

Each *CMS* includes 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 *CMS* securely over the Internet.

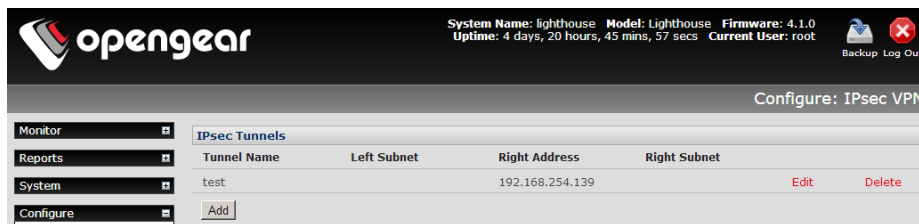
The administrator can establish an encrypted authenticated VPN connection between a virtual *CMS* or hardware *CMS* appliance on their central office network and a VPN gateway (such as Cisco router running *IOS IPsec* or an ACM5500 appliance) at a remote site.

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 their *CMS* (and its *Managed Console Servers* and their connected and managed devices)

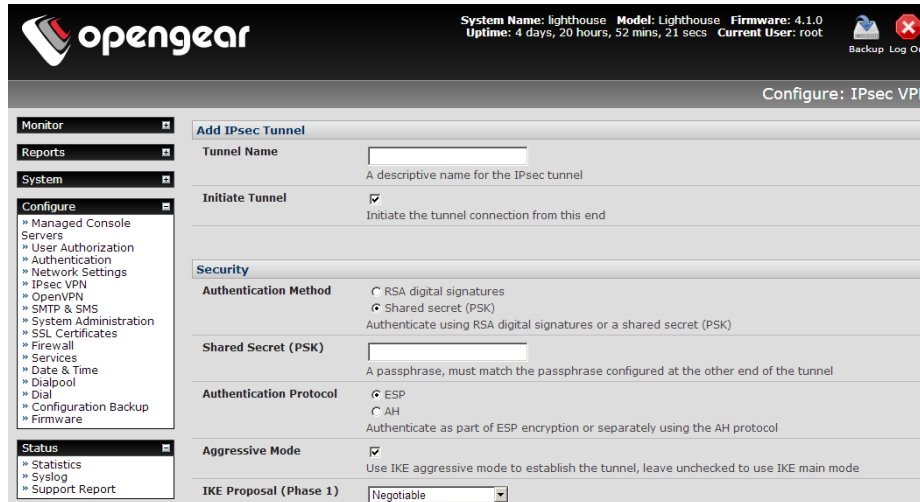
Configuration of IPsec is quite complex so Opengear provides a simple GUI interface for basic set up as described below. However for more detailed information on configuring Openswan IPsec at the command line and interconnecting with other IPsec VPN gateways and road warrior IPsec software refer <http://wiki.openswan.org>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

3.9.1 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*



- Select the **Authentication Method** to be used, either *RSA digital signatures* or a *Shared secret (PSK)*
 - If you select *RSA* you will be asked to [click here to generate keys](#). This will generate an RSA public key for the CMS (the *Left Public Key*). You will need to find out the key to be used on the remote gateway, then cut and paste it into the *Right Public Key*
 - If you select *Shared secret* you will need to enter a Pre-shared secret (PSK). The PSK must match the PSK configured at the other end of the tunnel
- In **Authentication Protocol** select the authentication protocol to be used. Either authenticate as part of *ESP* (Encapsulating Security Payload) encryption or separately using the *AH* (Authentication Header) protocol.
- Enter a **Left ID** and **Right ID**. This is the identifier that the Local host/gateway and remote host/gateway use for IPsec negotiation and authentication. Each ID must include an '@' and can include a fully qualified domain name preceded by '@' (e.g. *left@example.com*)
- Enter the public IP or DNS address of this Opengear VPN gateway (or if not an ACM5004-G or ACM5504-5-G-I enter the 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 Opengear VPN gateway is serving as a VPN gateway to a local subnet (e.g. the CMS 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 CMS 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 CMS 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 *CMS* (referred to as the Left or Local host) exactly matches the set up entered when configuring the Remote (Right) host/gateway or software client. Refer to the <http://www.opengear.com/faq.html> for details on configuring these remote ends

3.10 OpenVPN

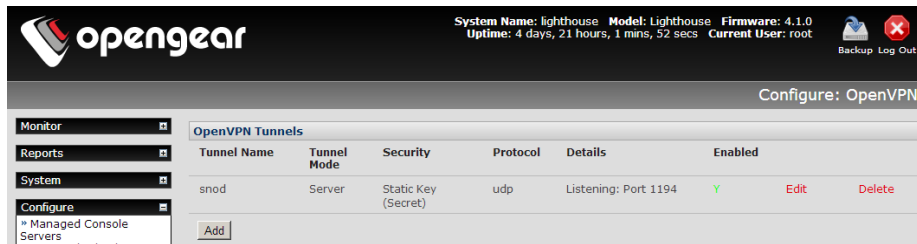
Each *CMS* includes OpenVPN which is based on TSL (Transport Layer Security) and SSL (Secure Socket Layer). With OpenVPN, it is easy to build cross-platform, point-to-point VPNs using x509 PKI (Public Key Infrastructure) or custom configuration files. The VPN allows multiple sites or remote administrators to access the *CMS* securely over the Internet.

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

Configuration of OpenVPN can be complex so Opengear 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>

3.10.1 Enable the OpenVPN

- Select **OpenVPN** on the **Serial & Networks** men
- 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*

The screenshot shows the 'Add OpenVPN Tunnel' configuration form. It contains the following fields and options:

- Tunnel Name:** A text input field containing 'NorthStOutlet-VPN'. Below it is the instruction: 'A descriptive name for the OpenVPN tunnel'.
- Device Driver:** A dropdown menu set to 'Tun - IP'. Below it is the instruction: 'Select the tap or tun driver to use.'
- Protocol:** A dropdown menu set to 'UDP'. Below it is the instruction: 'Use a UDP or TCP protocol'.
- Tunnel Mode:** A dropdown menu set to 'Client'. Below it is the instruction: 'Is this the Client or Server end of the tunnel.'
- Configuration Method:** A dropdown menu set to 'PKI (X.509 Certificates)'. Below it is the instruction: 'Authenticate using certificates or use a custom configuration'.
- Compression:** A checkbox that is checked. Below it is the instruction: '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 advanced CMS 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

Client Details	
Primary Server Address	<input type="text" value="192.168.250.106"/> The address of the first server.
Primary Server Port	<input type="text"/> The TCP/IP port of the first server. <i>Default is 1194.</i>
Secondary Server Address	<input type="text"/> The address of the second server (Optional).
Secondary Server Port	<input type="text"/>

3.10.2 Configure as Server or Client

- Complete the **Client Details** or **Server Details** depending on the Tunnel Mode selected.
 - If *Client* has been selected, the *Primary Server Address* will be the address of the OpenVPN Server.
 - 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	<input type="text" value="SouthStOutlet-VPN"/>	A descriptive name for the OpenVPN tunnel
Device Driver	<input type="text" value="Tun - IP"/> <input type="button" value="v"/>	Select the tap or tun driver to use.
Protocol	<input type="text" value="UDP"/> <input type="button" value="v"/>	Use a UDP or TCP protocol
Tunnel Mode	<input type="text" value="Server"/> <input type="button" value="v"/>	Is this the Client or Server end of the tunnel.
Configuration Method	<input type="text" value="PKI (X.509 Certificates)"/> <input type="button" value="v"/>	Authenticate using certificates or use a custom configuration
Compression	<input checked="" type="checkbox"/>	Enable or disable compression

Server Details

Local Port	<input type="text"/>	The TCP/IP port to listen on. <i>Default is 1194.</i>
IP Pool Network	<input type="text" value="10.100.0.0"/>	Network addresses to allocate.
IP Pool Netmask	<input type="text" value="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	<input type="text"/>	<input type="button" value="Browse..."/>	File is not custom	NorthStOutlet-VPN.conf
Root CA Certificate	<input type="text" value="ear\Testing\Certificates\ca.crt"/>	<input type="button" value="Browse..."/>	<input type="button" value="Upload"/>	No file available
Certificate File	<input type="text" value="ing\Certificates\acm-client.crt"/>	<input type="button" value="Browse..."/>	<input type="button" value="Upload"/>	No file available
Private Key File	<input type="text" value="ng\Certificates\acm-client.key"/>	<input type="button" value="Browse..."/>	<input type="button" value="Upload"/>	No file available
Diffie-Hellman File	<input type="text"/>	<input type="button" value="Browse..."/>	<input type="button" value="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	<input type="text"/>	<input type="button" value="Browse..."/>	File is not custom	NorthStOutlet-VPN.conf
Root CA Certificate	<input type="text"/>	<input type="button" value="Browse..."/>	<input type="button" value="Upload"/>	NorthStOutlet-VPN-ca.crt
Certificate File	<input type="text"/>	<input type="button" value="Browse..."/>	<input type="button" value="Upload"/>	NorthStOutlet-VPN-public.crt
Private Key File	<input type="text"/>	<input type="button" value="Browse..."/>	<input type="button" value="Upload"/>	NorthStOutlet-VPN-private.key
Diffie-Hellman File	<input type="text"/>	<input type="button" value="Browse..."/>	<input type="button" value="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	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

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

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

Edit OpenVPN Tunnel Details

Edit OpenVPN Tunnel Details

Tunnel Name	NorthStOutlet-VPN A descriptive name for the OpenVPN tunnel
Enabled	<input checked="" type="checkbox"/> 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	<input checked="" type="checkbox"/> Enable or disable compression

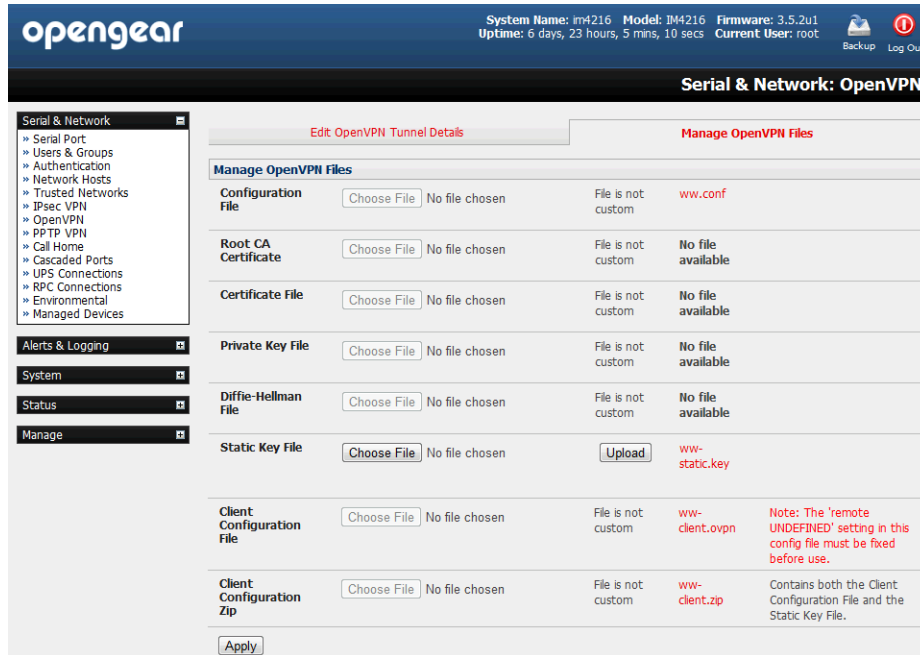
➤ Select **Statistics** on the **Status** menu to verify that the tunnel is operational.

Interfaces	Routes	Serial Ports	IP	ICMP	TCP
eth0					
Link encap:Ethernet HWaddr 00:10:A1:96:92:05 inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0 inet6 addr: fe80::210:a1ff:fe96:9205/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:2616 errors:0 dropped:0 overruns:0 frame:0 TX packets:1565 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 Interrupt:12 Memory:1fff8000-1fff80ff					
eth0:0					
Link encap:Ethernet HWaddr 00:10:A1:96:92:05 inet addr:192.168.250.111 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 inet6 addr: ::1/128 Scope:Host UP LOOPBACK RUNNING MTU:16436 Metric:1 RX packets:975 errors:0 dropped:0 overruns:0 frame:0 TX packets:975 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0					
tun0					
Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 inet addr:10.100.0.6 P-t-P:10.100.0.5 Mask:255.255.255.255 UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:100					

3.10.3 Windows OpenVPN Client and Server set up

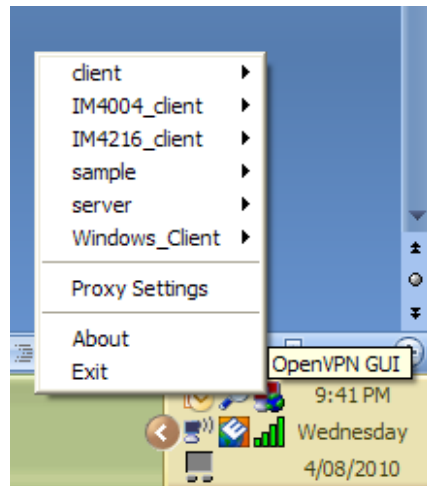
Windows does not come standard with any OpenVPN server or client. This section outlines the installation and configuration of a Windows OpenVPN client or a Windows OpenVPN server and setting up a VPN connection to a CMS.

CMS appliances will generate Windows client config automatically from the GUI – for **Pre-shared Secret (Static Key File)** configurations.



Alternately *OpenVPN GUI for Windows* software (which includes the standard OpenVPN package plus a Windows GUI) can be downloaded from <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 “.ovpn” 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: IM4216_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 IM4216_OpenVPN_Server

```

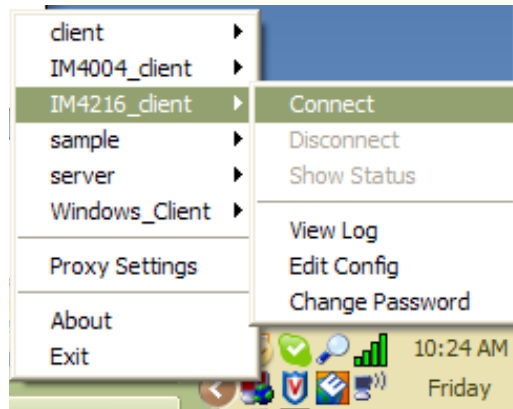
The Windows client/server configuration file options are:

Options	Description
#description:	This is a comment describing the configuration. Comment lines start with '#' 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 server'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, IM4216_client
- Click 'Connect' as shown below



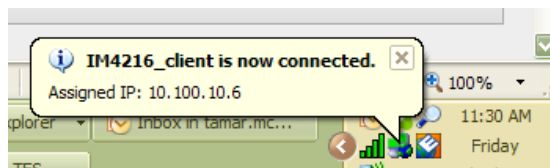
- The log file will be displayed as the connection is established

```

IM4216_client - Notepad
File Edit Format View Help
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@
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-B689-720FEC0B1F713}
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
Ln 1, Col 1

```

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



Note: An alternate 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



3.11 Firewall & Forwarding

CMS appliances have basic routing, NAT (Network Address Translation), packet filtering and port forwarding support on all network interfaces.

- **Network Forwarding** allows the network packets on one network interface to be forwarded to another network interface. So locally networked devices can IP connect through the *CMS* 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.
- When using IP Masquerading, devices on the external network cannot initiate connections to devices on the internal network. **Port Forwards** allows external users to connect to a specific port on the external interface of the *CMS* and be redirected to a specified internal address for a device on the internal network.
- With **Firewall Rules**, packet filtering inspects each packet passing through the firewall and accepts or rejects it based on user-defined rules.
- Then **Service Access Rules** can be set for connecting to the *CMS* itself

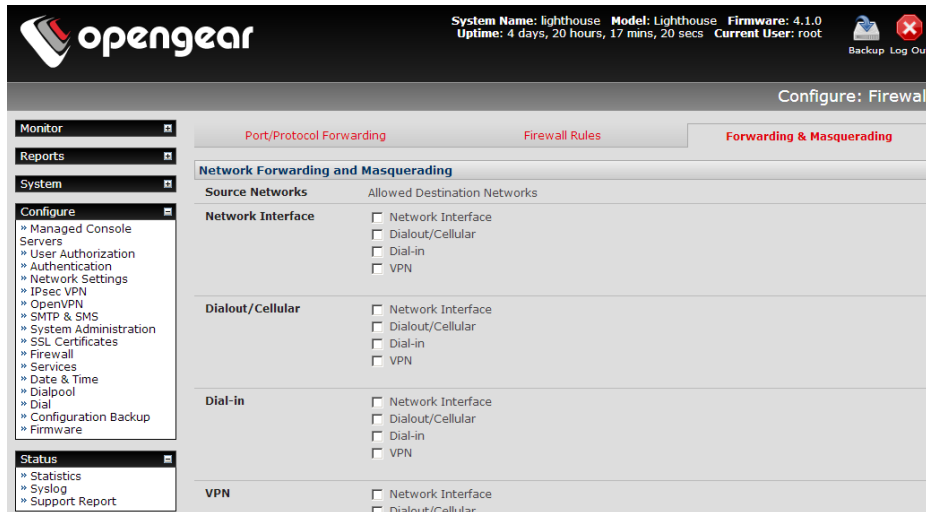
3.11.1 Configuring network forwarding and IP masquerading

To use a *CMS* 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 to be forwarded to another network interface. So locally networked devices can IP connect through the *CMS* 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 *CMS* models are configured so that they will not route traffic between networks. To use the *CMS* 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



➤ Find the **Source Network** to be routed, and then tick the relevant **Destination Network** to enable Forwarding. IP Masquerading is generally required if the CMS 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 CMS.

IP Masquerading performs Source Network Address Translation (SNAT) on outgoing packets, to make them appear like they've come from the CMS (rather than devices on the internal network). When response packets come back devices on the external network, the CMS will translate the packet address back to the internal IP, so that it is routed correctly. This allows the CMS to provide full outgoing connectivity for internal devices using a single IP Address on the external network.

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

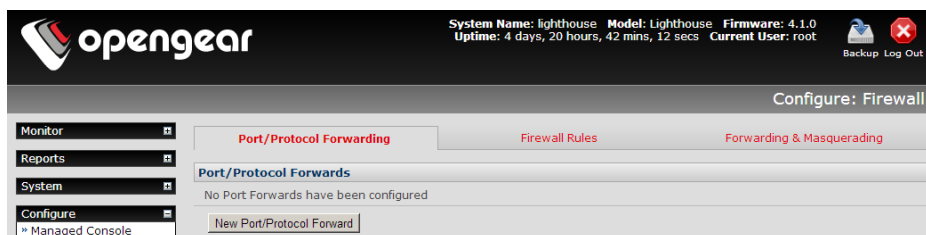
- Select **Forwarding & Masquerading** panel on the **System: Firewall** menu
- Check **Enable IP Masquerading (SNAT)** on the network interfaces where masquerading is to be enabled

Generally this masquerading would be applied to any interface that is connecting with a public network such as the Internet (e.g. for the CMS with an external modem the IP masquerading would be enabled on **Dialout/Cellular**)

3.11.3 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 CMS, and have the CMS redirect the data to a specified internal address and port range.



To setup a port/protocol 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)

Destination Address/Address Range: The destination IP address/address range to match. This may be left blank IP address ranges use the format *ip/netmask* (where netmask is in bits 1-32)

Input Port Range: The range of ports to forward to the destination IP. These will be the port(s) specified when accessing the port forward. These ports need not be the same as the output port range.

Protocol: The protocol of the data being forwarded. The options are *TCP* or *UDP* or "*TCP and UDP*" or *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 range of ports that the packets will be redirected to on the Output Address. Ranges use the format *start-finish*. Only valid for TCP and UDP protocols

3.11.4 Firewall rules

Firewall rules can be used to block or allow traffic through an interface based on port number, the source and/or destination IP address (range), the direction (ingress or egress) and the protocol. This can be used to allow custom on-box services, or block traffic based on policy.

To setup a firewall rule:

- Navigate to the **System: Firewall** page, and click on the **Firewall Rules** tab

- Click **New Firewall Rule**
- Fill in the following fields:
 - Name:** Name the rule. This name should describe the policy the firewall rule is being used to implement (e.g. *block ftp*, *Allow Tony*)
 - 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*

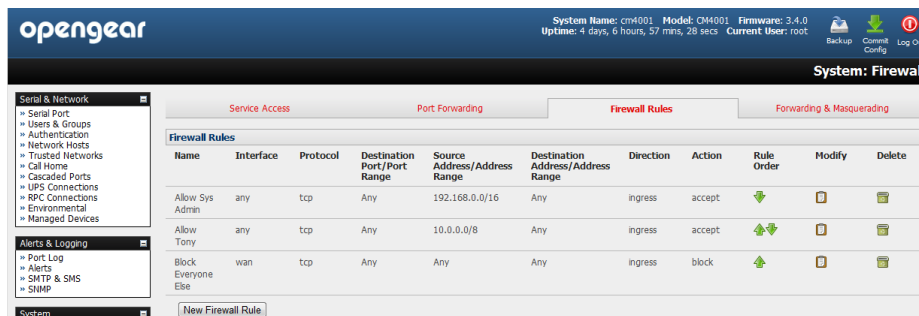
- Source MAC address Specify the source MAC address to be matched. This may be left blank for any. MAC addresses use the format XX:XX:XX:XX:XX:XX, where XX are hex digits
- Source Address Range: Specify the source IP address (or address range) to match. IP address ranges use the format ip/netmask (where 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* or “*TCP and UDP*” or *ICMP* or *ESP* or *GRE* or *Any*
- 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 all SSH traffic from leaving Dialout Interface, the following settings can be used:

- Interface: *Dialout/Cellular*
- 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*):

	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	<i>Any</i>	<i>Any</i>	<i>Network Interface</i>
Port Range	<i>Any</i>	<i>Any</i>	<i>Any</i>
Source MAC	<i>Any</i>	<i>Any</i>	<i>Any</i>
Source IP	<i>IP address of SysAdmin</i>	<i>IP address of Tony</i>	<i>Any</i>
Destination IP	<i>Any</i>	<i>Any</i>	<i>Any</i>
Protocol	<i>TCP</i>	<i>TCP</i>	<i>TCP</i>
Direction	<i>Ingress</i>	<i>Ingress</i>	<i>Ingress</i>
Action	<i>Accept</i>	<i>Accept</i>	<i>Block</i>



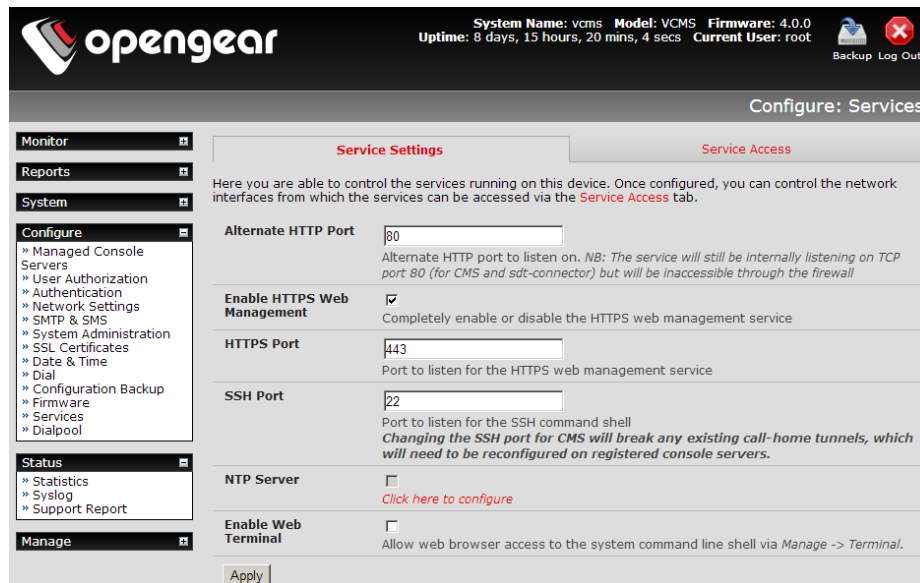
However if the **Rule Order** above was to be changed so the “*Block Everyone Else*” rule was second on the list then the traffic coming in over the *Network Interface* from *Tony* would be blocked.

3.12 Services and Service Access

The *Administrator* can access the *CMS* (and its *Managed Console Servers* and their connected and managed devices), using a range of access protocols/services. You can control the services running on the *CMS* and the network interfaces from which the services can be accessed.

To enable and/or configure a service:

- Select the **Service Settings** tab on the **Configure: Services** page



- Enable and configure basic services:

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

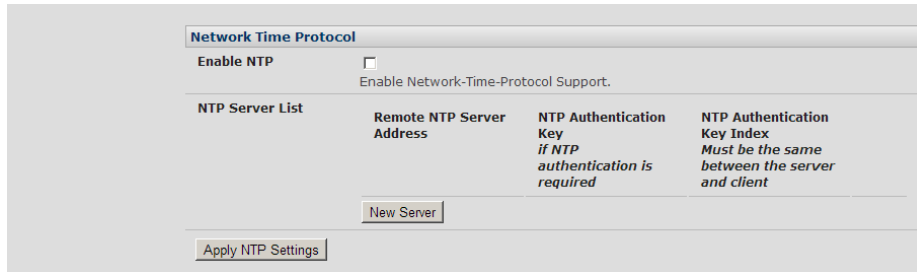
Alternate HTTP also enables you to configure an alternate HTTP port to listen on. However the HTTP service will continue internally listening on TCP port 80 (for *CMS* and *sdt-connector* communications) but will be inaccessible through the firewall.

HTTPS By default the HTTPS service is running and this service is enabled on all network interfaces. It is recommended that only HTTPS access be used if the *CMS* is to be managed over any public network (e.g. the Internet). This ensures the *Administrator* has secure browser access to all the menus on the *CMS*. The HTTPS service can be completely disabled (or re-enabled) by checking **HTTPS Web Management** and an alternate port specified (default port is 443).

SSH This service provides secure SSH access to the *CMS* and the SSH service is always running and by default is enabled on all interfaces. An alternate SSH port to listen on can be specified in **SSH command shell port** (default port is 22). However changing the SSH port for *CMS* will break any existing call-home tunnels, which will need to be reconfigured on registered *CMS*s.

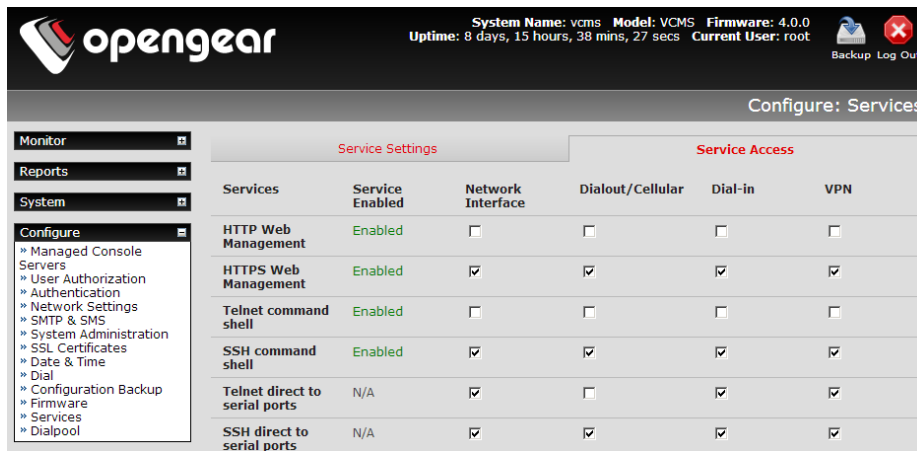
Web Terminal Checking **Enable Web Terminal** will allow web browser access to the system command line shell via **Manage: Terminal**

NTP Configuring NTP ensures the *CMS* clock is kept extremely accurate (once Internet connection has been established). Select the **Enable NTP** checkbox 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 NTP Settings**



To control the network interfaces from which the services can be accessed

- Select the **Service Access** tab on the **System: Services** page. This will display the services currently enabled for the *Lighthouse* appliance’s network interfaces.

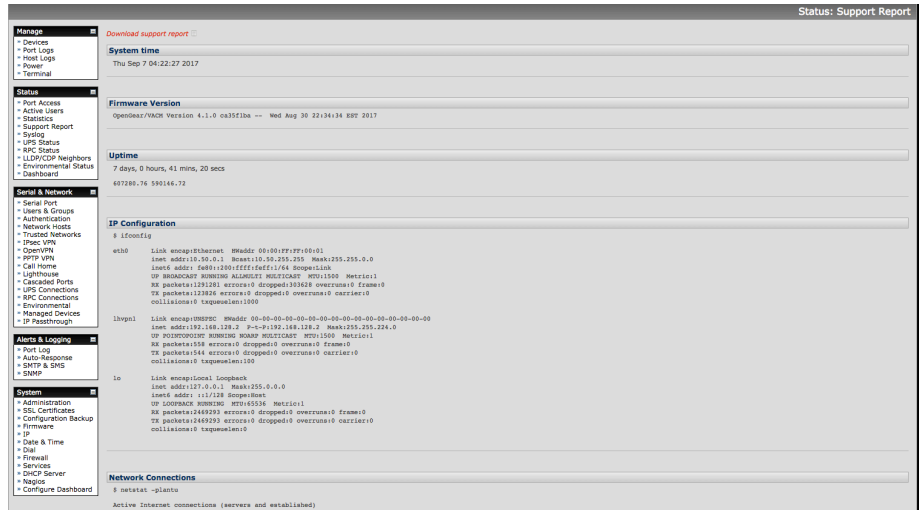


- Check/uncheck for each network which service access is to be enabled /disabled
- Click **Apply** to apply your services access selections

3.13 Support Report

The Support Report provides useful status information that will assist the Opengear technical support team to solve any problems you may experience with your *CMS*.

If you do experience a problem 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 to your email support request.



- Select **Status: Support Report** and you will be presented with a status snapshot
- Click the **Download support request link**.
- A PKZip file with a filename structured as follows — `console-server-hostname_yyyymmdd_support-report.zip` — is downloaded to the local system.

3.14 System Reset

The *Administrator* can reboot or reset the *CMS* to default settings.

A *soft* reset is effected by:

- Select **Reboot** in the **Configure: System Administration** menu to safely reboot your *Lighthouse* appliance. The *CMS* 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.
- Select **Config Erase** to erase all configurations and restore factory default settings. This setting requires a safe reboot. On reboot you will be prompted to enter and confirm a new root password before the device (UI and ssh) can be accessed
- Clicking **Apply**

Config Erase	<input type="checkbox"/>	Restore factory default settings (requires safe reboot).
Reboot	<input type="checkbox"/>	Safely reboot the device.
Shut Down	<input type="checkbox"/>	Safely shut down the device.
<input type="button" value="Apply"/>		

3.15 Syslog

The Linux System Logger in *CMS* maintains a record of all system messages and errors. The syslog record can be redirected to a remote Syslog Server:

- Select **Status: Syslog** and 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**

- **Local Log Level** enables you to limit the amount of Syslog information being logged by specifying event types to be logged
- 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
- Click **Apply**. The Syslog will then be represented with only those entries of the nominated event type and that actually include any specified pattern

3.16 Dialpool – centralized dial-out

The dialpool facility enables distributed modems to be built into a simple virtual modem pool, and accessed centrally from CMS using a browser. The modems in the dialpool are either serially connected to downstream console servers, or internally integrated within the downstream console servers (e.g. ACM5004-M). These downstream console servers are network connected to CMS, and can be located adjacent to the Lighthouse appliance or distributed regionally - or internationally.

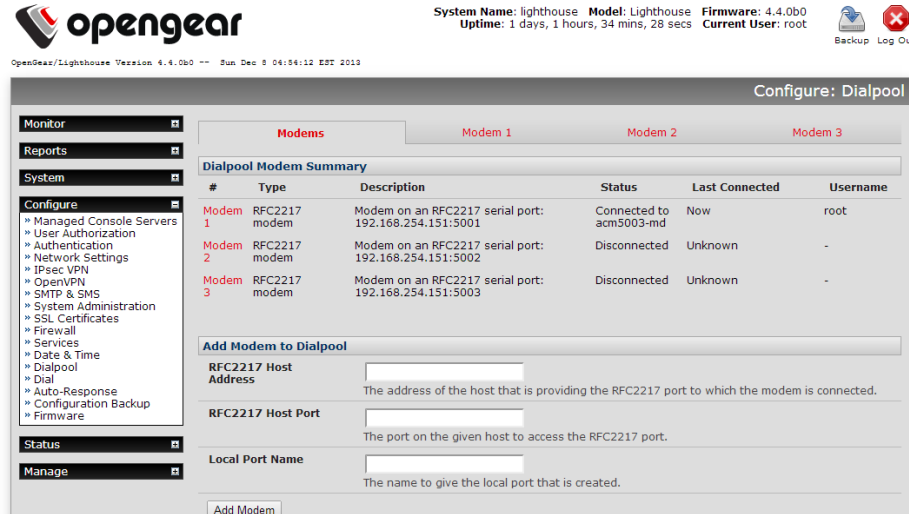
So dialpool provides a centralized dial-out capability. Admins can from anywhere use their browser to trigger a dial-out session and connect to the Managed Console Servers at remote sites out-of-band via analog modem connection – without the need to carry their own modem. Also by distributing the downstream console servers (that host the dialpool modem(s)) within the remote countries and regions, they can overcome international line quality issues from trying to use compressed PSTN lines and dialing globally.

Establishing the dialpool and initiating a dial-out connection is a simple process off:

- Establishing RFC2217 connections to the modem port(s) on downstream console servers that are IP connected to the Lighthouse CMS
- Adding each modem to the CMS dialpool and adding the connection details for the *Managed Console Server* that is to be dialed (e.g. out-dial phone #)
- Configuring the *Managed Console Server* for in-dial access
- Clicking *Dial*

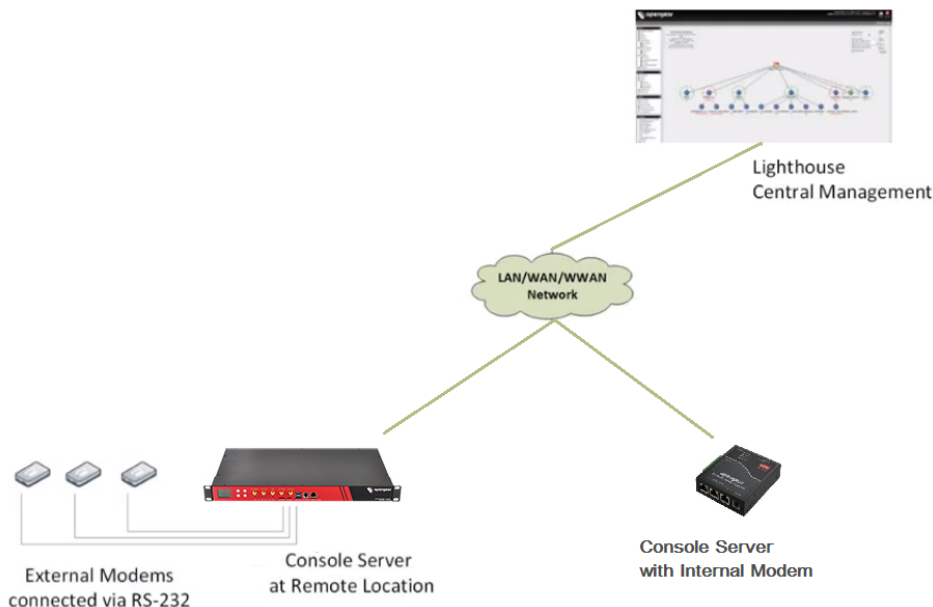
3.16.1 Dialpool setup

The **Configure: Dialpool** shows the modems in the dialpool. The Modems tab (V4.3 and later) presents a summary view with details about modem setup, type, current status information (**disconnected**, **dialling**, or **connected**) as well as the connection time and the user that initiated dialling. More details are provided within each of the individual modem tabs.



3.16.2 Add modems to the dialpool

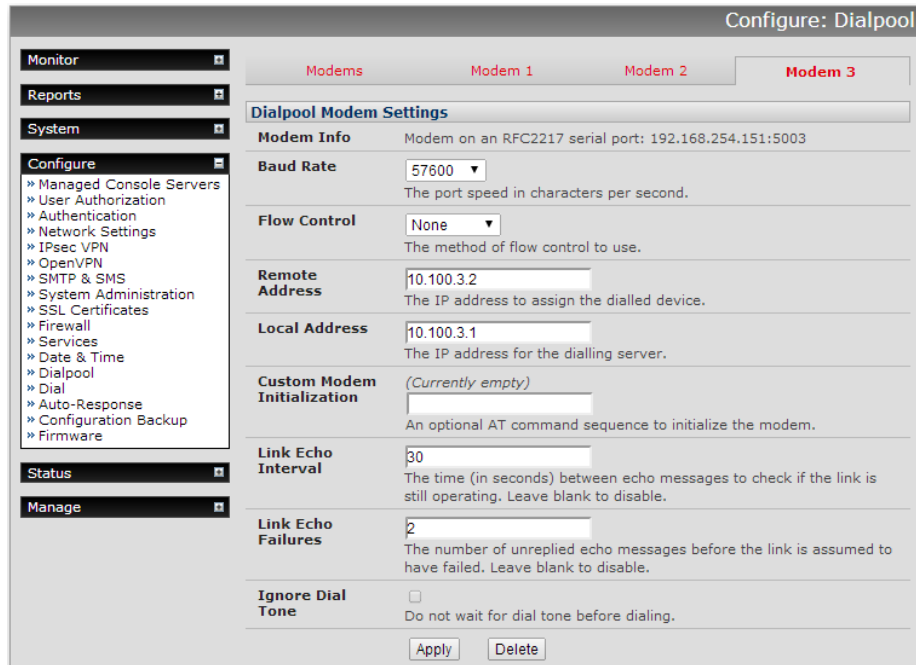
- To add a modem to the dialpool select **Configure: Dialpool** and go to the **Add Modem to Dialpool** section. Enter the **Host Address** (i.e. the IP address or Domain Name of the downstream console server with the modem), and the **RFC2217 Host Port** address of the modem (i.e. the tcp port # of the modem serial port e.g. 5011)



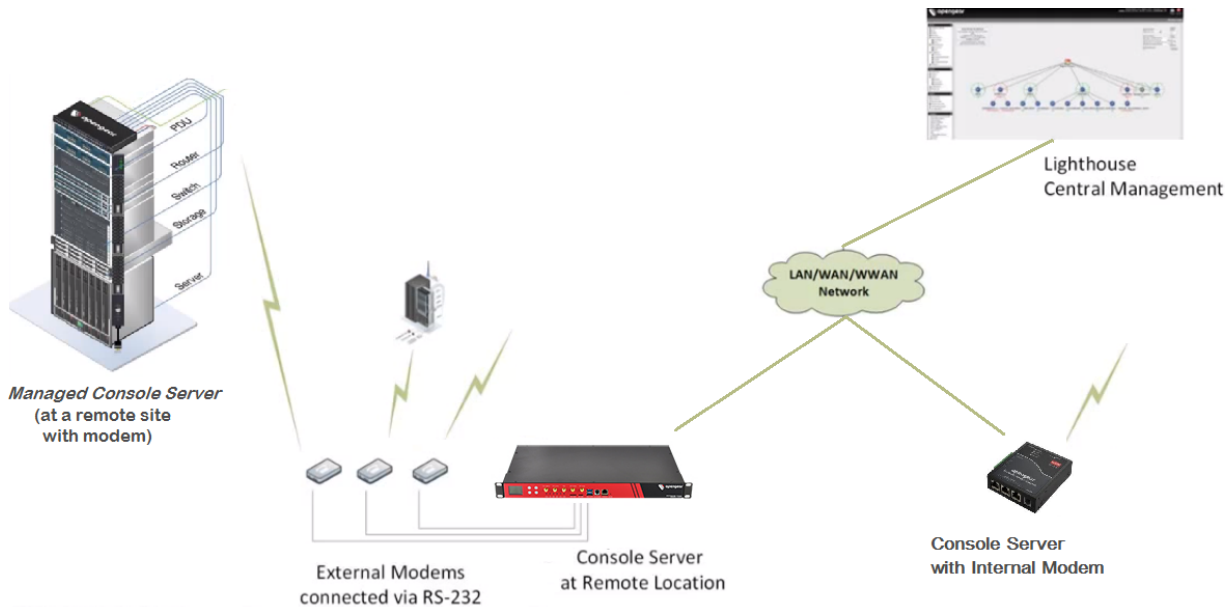
Note RFC2217 provides for *virtual* serial port connections, and the serial port with the modem must have this enabled. To enable this, when configuring the downstream console server, simply tick *RFC2217* as the *Console Server Setting* for the specific serial port in the *Serial&Network:Serial Ports* menu.

- Provide the modem a **Local Port Name** and click **Add Modem**

- A new tab will have been created on the **Configure: Dialpool page** for the modem you have added. Select this tab and set up for your out-dial settings



Once you have the modem pool set up for your out dial, you will need to set up the phone numbers etc of the modems on the *Managed Console Servers* you may wish to dial into for out of band management.



As you add new *Managed Console Servers*, or edit existing ones:

- In **Configure: Managed Console Servers** configure **Remote Dialin Setup** to allow it to be accessed and managed through a dial in connection

Remote Dialin Setup

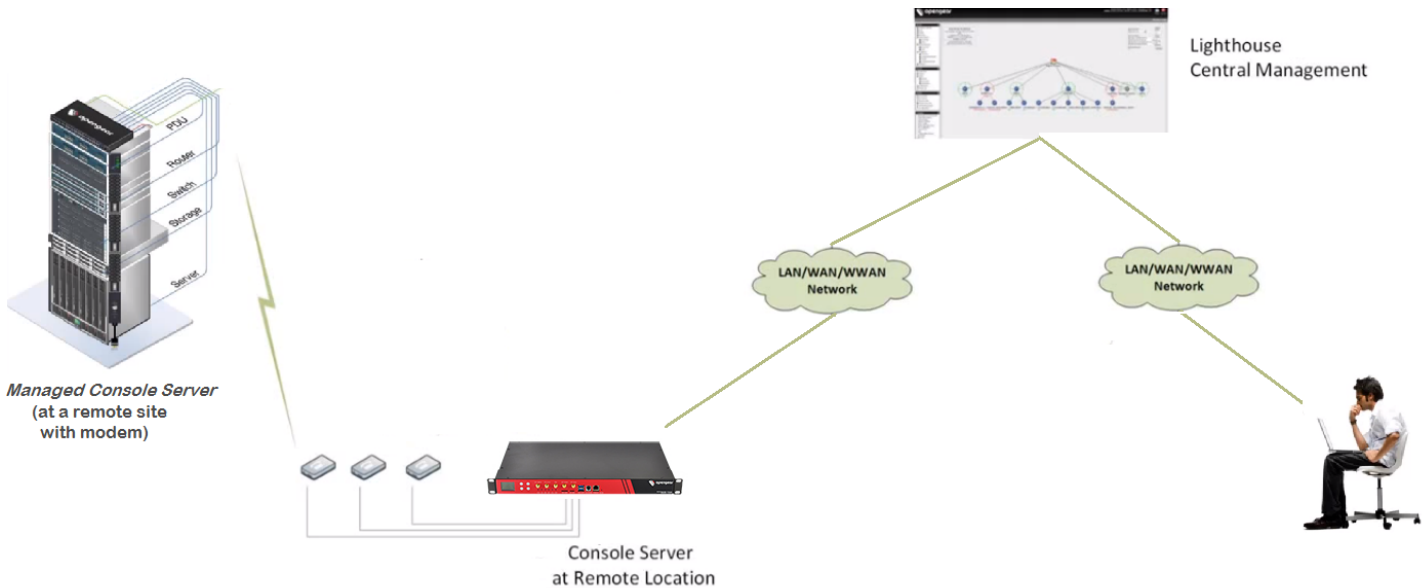
This section allows the managed console server to be configured for access and management through a dial connection on the server.

Phone Number	<input type="text"/>	The phone number to access the remote console server on.
Dialin Username	<input type="text"/>	The username for an account on the remote console server with dialin access.
Dialin Password	<input type="password"/>	The password for the given dialin account.
Call Home Address	<input type="text"/>	(optional) The server address that the remote console server is using as a call home tunnel.

- Enter the **Phone Number** to access the remote *Managed Console Server* on
- Enter the **Dialin Username** and **Password** for a user account on the *Managed Console Server* with dialin access

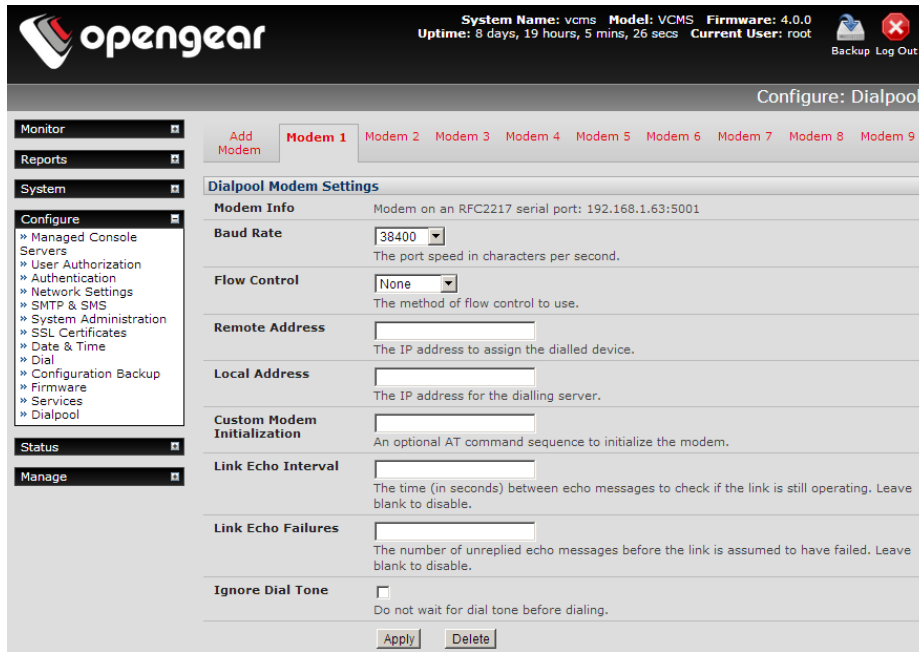
3.16.3 Dialing Managed Console Servers

There are two paths for accessing any of these configured *console servers* through the virtual dialpool.

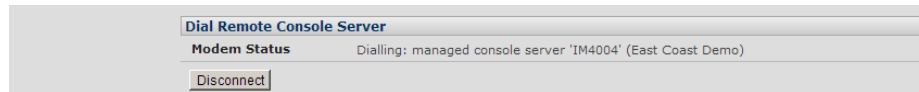


The **Configure: Dialpool** page allows you to manually select any modem from the pool and dial:

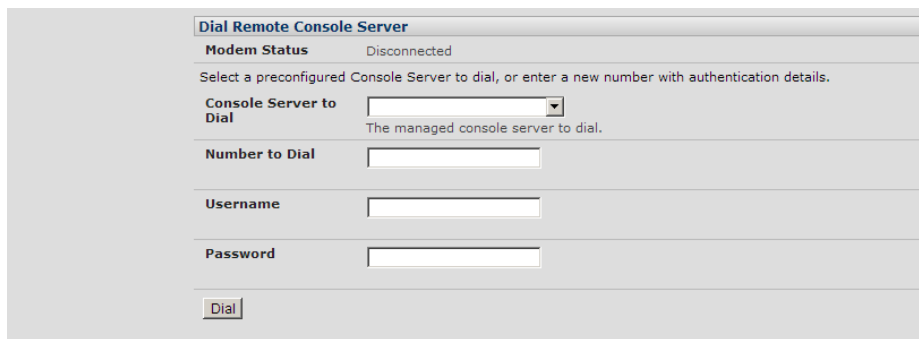
- Select **Configure: Dialpool** and select any modem in the virtual pool



- If the modem is already in use (**Dial Remote Console Server** reports *Modem Status: Dialling*), then simply browse for a free modem in the pool

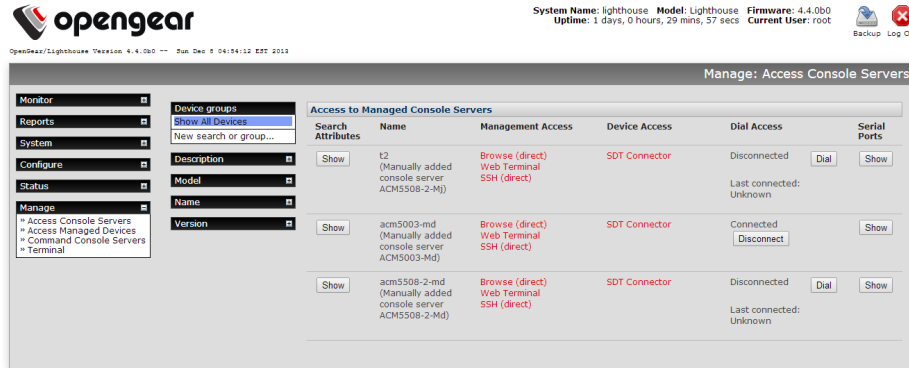


- Select a preconfigured **Console Server to Dial** from the drop down list, or enter a new **Number to Dial** with authentication details
- Hit **Dial**



- You will now be dial connected to the *Managed Console Server* and can access it and its Managed Devices through the **Manage: Access Console Servers** page

Alternately, with software V4.3 and later, the **Manage: Access Console Servers** page includes an extra column for dial access which will automatically select a free modem from the pool and dial:



- Select **Manage: Access Console Servers**. This will show, for each managed console server, if it has been setup for dial access, and if so, the current status.
- Buttons are provided to either dial the console server (if disconnected), or disconnect a currently dialed connection.

3.16.4 Dialpool health monitoring

The dialpool health test (V4.3 and later) is configured through the Auto-Response interface. A new test type, "Dialpool Health", is available, which gives the following options to test connectivity to managed console servers and/or dialpool modems:

- **Test Managed Console Servers:** if managed console servers (with dial access configured) should be regularly tested, as per the specified parameters.
- **Threshold:** the number of hours/days/weeks/months that a managed console server must not have been accessed for, before the health test will attempt to dial it.
- **Max Modem Count:** the maximum number of modems to use in parallel to test dial managed console servers. This can be used to ensure that some modems are always available for use.
- **Test Dialpool Modems:** if dialpool modems should be tested if they have not been used within the specified timeframe.
- **Threshold:** the number of hours/days/weeks/months that a modem must not have been used before it will be tested by the health check.

Once configured, the health can be used with the standard Auto Response actions e.g. it could send a notification email when a modem or console server fails to connect.

When using an action for the dial health test Auto-Response, there are some extra custom variables that can be used:

- **DIAL_DEVICE:** the type of device that has failed, either 'modem' or 'node'.
- **DIAL_NAME:** the name of the device that has failed, or the modem number if a modem.
- **DIAL_LASTTIME:** The last time this device was successfully connected.
- **DIAL_FAILURES:** The number of repeated failures this device has had since last connect.

An example alert message could be:

\$TIMESTAMP: Dial health test failed connection to \$DIAL_DEVICE \$DIAL_NAME. Last connected time was \$DIAL_LASTTIME. Has failed \$DIAL_FAILURES times.

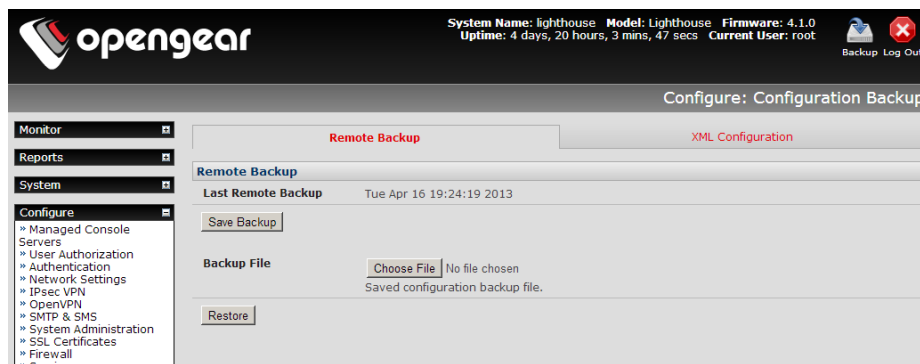
3.17 Configuration Backup

It is recommended that you back up the *CMS* 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



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

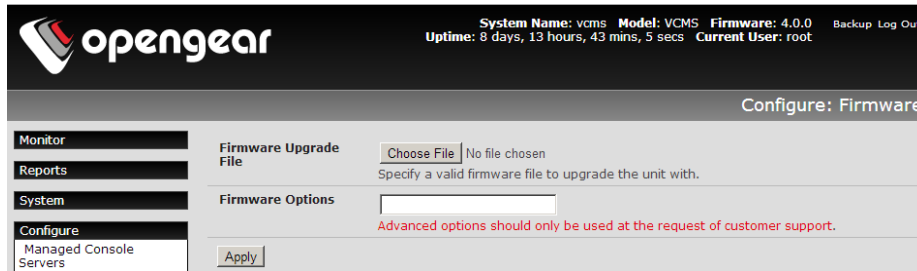
To restore a remote backup:

- 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*

3.18 Upgrade Firmware

Before upgrading, you should ascertain if you are already running the most current firmware in your *Lighthouse* appliance. Your *CMS* will not allow you to upgrade to the same or an earlier version.

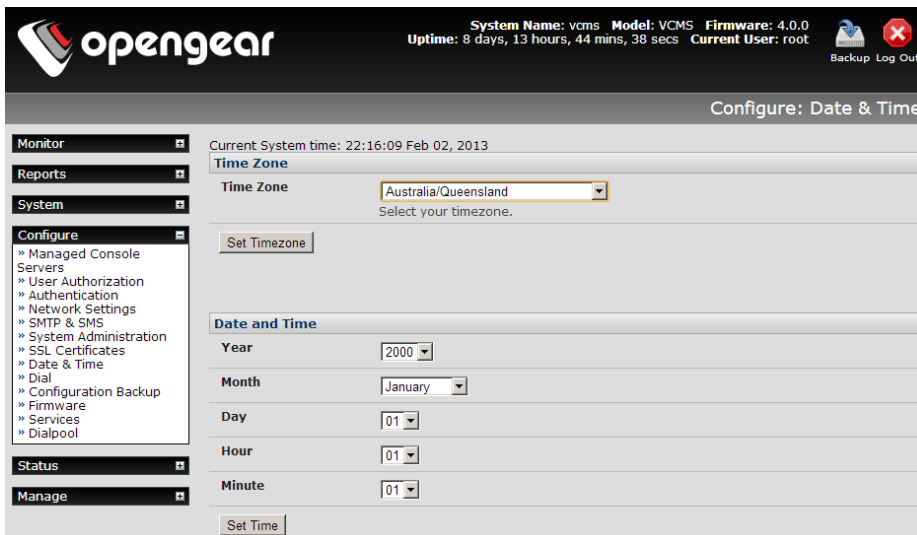
- The *Firmware* version is displayed in the header of each page or you can select **Configure: Support Report** and note the *Firmware Version* listed there
- The Lighthouse upgrade files (*.bin) are available from <http://www.opengear.com/firmware/>. Which upgrade file you use also depends on your solution.
 - For Lighthouse VM (VMware and Linux KVM) use *lighthouse-x.y.z-vm.bin* (eg. lighthouse-4.1.0u1-vm.bin)
 - For Lighthouse Enterprise and Lighthouse Standard use *lighthouse-x.y.z-hw.bin*
- Save this downloaded firmware image file on to a system on the same subnet as the *CMS*



- Also download and read the *release_notes.txt* for the latest information
- To upload the firmware image file to your CMS select **Configure: Firmware**
 - **Browse** the local PC and locate the downloaded file
 - Click **Apply** and the *Lighthouse* 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 CMS will have retained all its pre-upgrade configuration information

3.19 Configure Date and Time

It is recommended that you set the local Date and Time in the CMS as soon as it is configured. Many of the CMS logging features 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



- Select the **Configure: Date & Time** menu option
- Set your appropriate region/locality in the **Time Zone** selection box (not UTP) and click **Apply**
- Manually set the **Year, Month, Day, Hour** and **Minute** using the **Date** and **Time** selection boxes, then click **Apply**

Alternately, the CMS can synchronize its system time with a remote time server using the Network Time Protocol (NTP). Configuring the NTP time server ensures that the CMS clock will be accurate soon after the Internet connection is established. To set the system time using NTP:.

- Select the **Enable NTP** checkbox in the **Network Time Protocol** section

- 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 NTP Settings**

3.20 Key Exchange

The *CMS* automatically generates the SSH keys used to communicate with each of its *Managed Console Servers*.

However, you can additionally generate or manually enter RSA or DSA key pairs and SSH Authorized keys that will be used for other SSH connections with the *CMS*.

- Select **Configure: System Administration**
- Check **Generate SSH keys automatically** and click **Apply**

Next you must select whether to generate keys using RSA and/or DSA (and if unsure check only **RSA Keys**). Generating each set of keys will require approximately two minutes and the new keys will destroy any old keys of that type that may have previously been uploaded. To generate keys:

- Select **RSA Keys** and/or **DSA Keys**
- Click **Apply**
- Once the new keys have been successfully generated simply click **here** to return

Alternately if you have a RSA or DSA key pair you can manually upload them to the CMS:

- Select **Configure: System Administration** on the CMS
- 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**

3.21 Console Gateway

CMS provides a single point of access for multiple console servers, and the *Console Gateway* feature allows command line access to any of the devices serially connected to the console servers through a single IP address. This feature requires Lighthouse CMS software V4.4 (and later) and console server firmware V3.9.0u3 (and later).

3.21.1 Configuring the Console Gateway

To enable the *Console Gateway* functionality on a *Managed Console Server*, the number of serial ports it has must be specified.

- This is either done at the time of adding new *Managed Console Servers* or by editing existing *Managed Console Servers*. As described in Section 3.4.1 this involves specifying the number of serial ports to be proxied (which generally would be all the serial ports on the console server)

The screenshot shows a configuration window titled 'Configure' with a sidebar menu on the left. The sidebar menu includes: Managed Console Servers, User Authorization, Authentication, Network Settings, SMTP & SMS, System Administration, SSL Certificates, Date & Time, Dial, Configuration Backup, Firmware, Services, and Dialpool. The main content area is titled 'Serial Port Proxy' and contains three input fields with labels and descriptions:

Number of Serial Ports	4	The number of serial ports on the managed console server to proxy via CMS. Leave blank to disable all serial proxy access.
RFC2217 Proxy Port Base	0	TCP port base for RFC2217 access via CMS. Leave blank to disable RFC2217 serial proxy access.
Raw TCP Proxy Port Base	0	TCP port base for Raw TCP access via CMS. Leave blank to disable Raw TCP serial proxy access.

- With the *Console Gateway* the menu generation and authorization decisions take place on Lighthouse, so it is important to keep the console servers synchronized with the Lighthouse after configuration changes. This can be done by going to the Managed Console Servers page, selecting the console server, and clicking **Retrieve Managed Devices**. Failure to do so may mean that port labels and user authorization become inconsistent.

When the Lighthouse retrieves the config from the device, it gets a list of all of the serial ports, their labels, and the users and groups that have access to them. This is what is used to generate the menus, and to do the authorization.

3.21.2 Console Gateway access

To use this functionality, use SSH to connect to the Lighthouse appliance, with the username format `username:serial` This will connect to the Lighthouse, and present a list of console servers that the user has access to.

Once the user selects a console server, they are presented with a list of console ports the user has access to. When one is selected, the user is connected to that port.

For faster access, there are some shortcuts that can be used in the username format that can give more specific lists of serial ports, or direct access without a menu:

`username:console_server_name`

When a valid console server name is specified, a list of console ports that the user has access to on that console

server will be presented. If they do not have access to that console server, the connection will fail.

username:console_server_name:port_name

When a valid console server name and port name are specified, and the user has access to that console server and port, the user will be directly connected to that port. If they do not have access to that port, the connection will fail.

username:port_name

When a valid port name is specified, the user will be connected to the first console server port with that port name found. If the user does not have access to that port, the connection will fail.

Note The console server names and port names are not case sensitive

3.21.3 Authentication and Authorization

Lighthouse supports both local and remote authentication including Radius, TACACS, LDAP (Active Directory), and Kerberos

Local Authentication

When local authentication is used, users are retrieved from the managed console servers and are given authorization and passwords via the User Authorization page on the Lighthouse. The user will have access to console servers that the users were retrieved from. For example, if the user "joe" exists on three console servers, then he will have access to those three console servers, and any ports on those console servers that he has been given access to (via the console server's User page)

Remote Authentication

Remote authentication can be used in two ways: Authentication only or Authentication and Authorization.

When the remote authentication service is used for authentication only, the users must exist on the console servers that they have access to, and be configured in the manner described above in the "Local Authentication" section. Kerberos can only be used in this mode.

When the remote authentication service is being used for authentication and authorization, minimal configuration of the console servers is needed to provide many users access to the ports. This is enabled by ticking the "Use Remote Groups" box on the Authentication page.

Radius, LDAP and TACACS can be configured to pass a list of groups back as part of an authentication operation. If any of these groups exist on the managed console servers, then the remote users will have access to those console servers, and the ports specified in the group definition on the console server.

For example, if on each console server, there is a NetworkGroup group configured that has access to a number of ports on each server (e.g. switches and routers), then any remote user that has NetworkGroup as a group will get access to those ports.

TACACS can also be further configured to provide a list of console servers and ports that the user has access to. Below is a snippet from a *tac_plus* configuration:

```
user = tu1 {
  service = raccess {
    port2 = acm5003/port02
    port3 = acm5003/port03
    port4 = kcs6104/port02
    port5 = kcs6104/port03
    port6 = cm4116/port01
    port7 = cm4116/port02
    port8 = cm4116/port03
    port9 = cm4116/port04
```

```
port10 = kcs6104/port04
port11 = acm5003/port01
port12 = img4216-25/port01
port13 = img4216-25/port02
port14 = img4004-5/port01
port15 = img4004-5/port02
port16 = cm4001/port01
port17 = imx4216/port01
port18 = imx4216/port03
port19 = im4248-34/port01
port20 = acm550x/port04
priv-lvl = 4
}
}
```

This snippet shows a number of console servers and ports that the user tu1 has access to. The important parts of a config line are as follows:

```
port2 = acm5003/port02
port2 <- this is an access list index, it is not related to a specific port number on the console server
acm5004 <- this is the name of the console server in Lighthouse
port02 <- this is the number of the port on the console server (port numbers start at port01)
```

This configuration syntax allows full access configuration to be done at the central point, rather than requiring extra groups to be configured on the console server. Please note that this only works for the Serial port concentrator functionality - the WebUI access console server page does not support this yet.

ACCESSING MANAGED CONSOLE SERVERS & DEVICES

The *CMS* provides a simple way to monitor and access *Managed Console Servers* and *Devices* using a single sign-on. It also provides a selection of paths through which network engineers and system administrators can access and manage their *Managed Console Servers* and attached *Managed Devices* and serial ports.

These include browser, web terminal, SSH and SDT proxy connection facilities. This chapter covers these access paths, and covers some batch command facilities reconfiguring *Managed Console Servers*.

4.1 Viewing Managed Console Servers & Devices

4.1.1 Viewing Managed Console Servers

The **Manage: Access Console Server** screen provides a search and filter-by-attribute tool for accessing and managing groups of *Managed Console Servers*.

- Click **Manage: Access Console Servers**. The *console servers* (and the *Managed Devices* and serial ports) that the current user has access to are listed under *Access to Managed Console Servers*

Note If the current *CMS* user has 'user' or 'admin' group access on a console server, they are deemed to have access to that console server

The screenshot shows the OpenGear CMS interface. At the top, the system name is 'oglh-emea', model is 'Lighthouse VM', and firmware is '4.5.4'. The uptime is '2 days, 0 hours, 38 mins, 34 secs' and the current user is 'root'. There are 'Backup' and 'Log Out' buttons. The main content area is titled 'Manage: Access Console Servers' and contains a table of console servers. The table has columns for 'Search Attributes', 'Name', 'Status', 'Management Access', 'Device Access', and 'Serial Ports'. Three servers are listed: 'demo-im', 'dev-wcm', and 'VACM-801'. Each row has a 'Show' button in the 'Search Attributes' column. The 'demo-im' server is 'Connected - Main', 'dev-wcm' is 'Connected - Main', and 'VACM-801' is 'Not Connected'. The left sidebar shows a navigation menu with categories like Monitor, Reports, System, and Configure. The 'Configure' menu is expanded to show 'Managed Console Servers'.

Search Attributes	Name	Status	Management Access	Device Access	Serial Ports
Show	demo-im (Demo IM7232-2-DAC-LR)	Connected - Main	Browse Web Terminal SSH	SDT Connector	Show
Show	dev-wcm (Dev ACM5504-5-G-W-I)	Connected - Main	Browse Web Terminal SSH	SDT Connector	Show
Show	VACM-801 (mess VACM #801 (Call Home))	Not Connected	Browse Web Terminal SSH	SDT Connector	Show

- The *Managed Console Servers* displayed can be filtered by attribute (Description, Location, Model, Names etc)
- Click **Show** in the **Search Attributes** column of any particular *Managed Console Server* to view and edit the attributes

System Name: oghl-emea **Model:** Lighthouse VM **Firmware:** 4.5.4
Uptime: 2 days, 0 hours, 38 mins, 34 secs **Current User:** root

Manage: Access Console Servers

Search Attributes	Name	Status	Management Access	Device Access	Serial Ports
Hide	demo-im (Demo IM7232-2-DAC-LR)	Connected - Main	Browse Web Terminal SSH	SDT Connector	Show

Access to Managed Console Servers

Description: Demo IM7232-2-DAC-LR
 Model: IM7232-2-DAC-LR
 Name: demo-im
 Version: 3.15.1
 Location: Cambridge
 Rack: CBLAB

- Click **Show** in the **Serial Ports** column of any particular *Managed Console Server* to view all devices attached to the serial ports of that *Managed Console Server*

System Name: oghl-emea **Model:** Lighthouse VM **Firmware:** 4.5.4
Uptime: 2 days, 0 hours, 38 mins, 34 secs **Current User:** root

Manage: Access Console Servers

Search Attributes	Name	Status	Management Access	Device Access	Serial Ports
Show	demo-im (Demo IM7232-2-DAC-LR)	Connected - Main	Browse Web Terminal SSH	SDT Connector	Hide Port 1 (Switch) Web Terminal Direct SSH Link Port 2 (Router) Web Terminal Direct SSH Link Port 3 (UPS) RFC 2217; 28202 Raw TCP; 28222 Port 4 (PDU) Web Terminal Direct SSH Link Port 5 (Office Switch) Web Terminal Direct SSH Link Port 6 No access configured. Port 7 No access configured. Port 8 (EMD)

4.1.2 Viewing Managed Devices

Managed Devices can also be viewed on the **Manage: Access Managed Devices** screen – which also provides filtering-by-attribute tool.

- Click **Manage: Access Managed Devices** and the *Managed Devices* displayed can be filtered by attribute (eg Console Server Description/ Location/ Model/Names etc or if is it a UPS or EMD device etc).

System Name: oghl-emea **Model:** Lighthouse VM **Firmware:** 4.5.4
Uptime: 2 days, 0 hours, 41 mins, 38 secs **Current User:** root

Manage: Access Managed Devices

Search Attributes	Name	Console Server	Managed Device Details
Show	Router (Cisco Router)	demo-im (Demo IM7232-2-DAC-LR)	Port 2 (Router) Web Terminal Direct SSH Link RFC 2217 Raw TCP RPC; PDU
Show	Router (Simulated Router)	VACM-801 (mass VACM #801 (Call Home))	Port 1 (Router) Web Terminal Direct SSH Link RPC; PDU
Show	Router (Simulated Router)	VACM-802 (mass VACM #802 (Call Home))	Port 1 (Router) Web Terminal Direct SSH Link RPC; PDU
Show	Router (Simulated Router)	VACM-803 (mass VACM #803 (Call Home))	Port 1 (Router) Web Terminal Direct SSH Link RPC; PDU
Show	Router (Simulated Router)	VACM-804 (mass VACM #804 (Call Home))	Port 1 (Router) Web Terminal Direct SSH Link RPC; PDU
Show	Router (Simulated Router)	VACM-805 (mass VACM #805 (Call Home))	Port 1 (Router) Web Terminal Direct SSH Link RPC; PDU
Show	Router (Simulated Router)	VACM-806 (mass VACM #806 (Call Home))	Port 1 (Router) Web Terminal Direct SSH Link RPC; PDU

Access to Managed Devices

Device groups: Show All Devices, Cisco, New search or group...

Console Server Description: Router (Cisco Router)

Console Server Location: Router (Simulated Router)

Console Server Model: Router (Simulated Router)

Console Server Name: Router (Simulated Router)

Console Server Rack: Router (Simulated Router)

Console Server Version: Router (Simulated Router)

Description: Router (Simulated Router)

Is EMD: Router (Simulated Router)

Name: EMD (1), Test (1), PDU (11), Switch (11), Router (11), Windows Server (1), Linux Server (1)

4.2 Accessing Managed Console Servers & Devices

4.2.1 Accessing Managed Console Servers

The **Management Access** column on the **Manage: Access Console Server** screen presents a selection of access paths to the *Managed Console Servers*:

- Click **Browse** to connect to the *Managed Console Server's* web UI. This connection is proxied via CMS, so the *console server* is still accessible even if firewalled, failed over to a private connection or otherwise inaccessible from the WAN. When browsing via a proxied connection, the following message is display in the Web UI header:

“This Console Server is being accessed via CMS **Click here to return to CMS**”



- Click **Web Terminal** to connect to the *Managed Console Server's* command line. The Web Terminal service uses AJAX to enable the web browser to connect through the CMS to the *Managed Console Server* using HTTPS as a terminal.

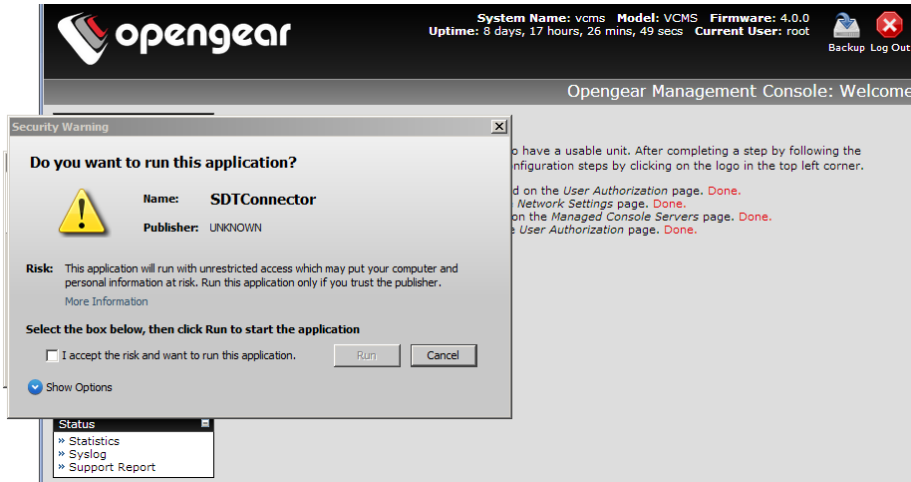


- Click **SSH** to establish an SSH link to the *Managed Console Server*. You will need to set up URL handlers for the `ssh://` links. The procedure here depends on the SSH client software and on the operating system you're using (Window, Ubuntu etc)

4.2.2 Accessing Managed Devices

The **Device Access** column on the **Manage: Access Console Server** screen presents a selection of access paths to the *Managed Devices*. These paths are also accessible from **Manage: Access Managed Devices** screen.

- Click **SDTConnector**. This will download a configured *SDT Connector* applet to your client PC and connect to the console server.



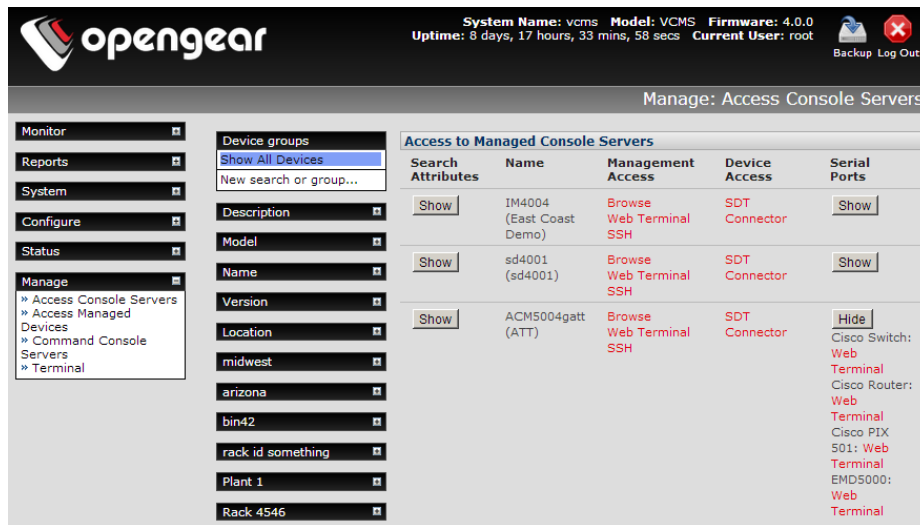
This also launches a command line shell session through the *SDT Connector* connection to the console server. As with Management Console connections, this connection is proxied via *CMS*.

The *SDT Connector* uses the credentials of the current user to connect to the console server. The *Managed Devices* and hosts that the current user has access to are retrieved, and displayed in the left hand column. For each host, connection buttons for the services the current user is permitted to access are available in the right hand Services pane. Click a service's button to launch a connection to it via *CMS*

Note When you click Connect it opens SDT Connector and launches a shell to the console server. This is exactly the same as when you click Connect for the "Command Line Shell" service on Monitor: Services screen as described in Chapter 6

Similarly the **Serial Ports** column of any particular *Managed Console Server* on the **Manage: Access Console Server** screen presents are a selection of access paths to serially attached devices.

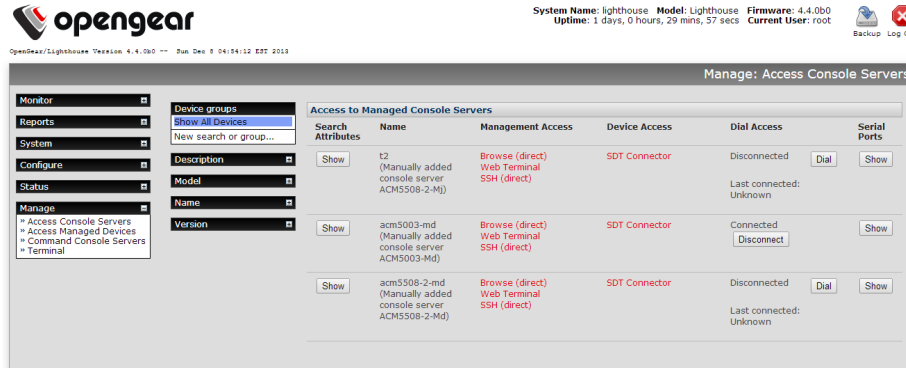
- Click **Show** to view all devices attached to the serial ports of that *Managed Console Server*



- Where configured you can then click **Web Terminal** to connect to the device on the remote *Managed Console Server's* serial port - as a terminal

4.2.3 Dialing Managed Console Servers

With firmware V4.3 and later the Access Managed Console Servers page includes an extra column for dial access. For details refer to the Dialpool section in Chapter 3.



4.3 Batch or Bulk Control of Managed Console Servers

CMS can queue commands to run on one or more *Managed Console Servers*. So network engineers can automate remote firmware upgrades, and system administrators can lock out nominated user access to specified sets of sites.

There are two systems to send commands to managed console servers en masse. The *node-** CLI commands execute commands directly via SSH and offers maximum flexibility and functionality for advanced users. The *Command Console Servers* UI offers simplified ease of use but limited functionality, scheduling and executing commands via the Nagios subsystem.

4.3.1 node-command Bulk CLI Command

The *node-command* CLI tool is used to run commands on managed console servers, allowing administrators to easily run a single CLI command in bulk, on all or on a range of their console server deployment.

Note To run *node-* commands, you must be authorized as an *admin* group user

Getting Started

To get started with any of the *node-* tools, you can get quick information on how to use it from the command line:

```
node-command --help
```

To see a list of all the registered console servers that the tool can operate on:

```
node-command --list-nodes
```

Selecting Console Servers

There are a number of ways to select console servers (aka “nodes”) as targets on which to run a command, listed below. These can be used multiple times, or together, to select a range of console servers:

Select individually by name, address, Call Home address or numeric config index (as per `--list-nodes` output):

```
node-command --node-name BNE-R01-IM4248
node-command --node-address 192.168.0.33:22
node-command --node-address localhost:40121
node-command --node-index 17
```

Select all:

```
node-command --all
```

Select multiple using regular expression pattern matches or inverse pattern match, against console server attributes defined under **Manage: Access Console Servers** UI:

```
node-command --select-match 'Model=ACM.*'
node-command --deselect-match 'Version=3.11.1'
```

Running Commands

Once console servers have been selected, the commands to be run for each can be given. These are run on each managed console server, in parallel. This command can be any command you can run from the console server CLI, commands are run as *root*.

For example to check the version on the first three configured console servers:

```
node-command --node-index 1 --node-index 2 --node-index 3 cat /etc/version
```

Note When using non-trivial selection arguments, you can check which target console servers have been selected an initial pass specifying `--list-nodes` rather than the final command

Copying Files

`node-copy` is a variation of `node-command` that can be used to copy a file to selected managed console servers, by specifying a source and destination, e.g. to copy a Cisco IOS system image that has been pre-copied onto the Lighthouse CMS's `/var/nvlog` directory, to all managed IM family console servers' TFTP server:

```
node-copy --select-match 'Model=IM.*' --source-file /var/nvlog/ios.img \
/var/tmp/usbdisk/tftpboot/
```

Output Format: CLI

The command outputs the result of each command run on each remote console server. For example, the example `node-command` from the Running Commands section gives the following result:

```
node-command --node-index 1 --node-index 2 --node-index 3 cat /etc/version
== node-command ID 2014-03-12T05:10:29.360164_29534 ==
15:10:29 [SUCCESS] BNE-R01-IM4248 10.10.0.1:22
OpenGear/IM42xx Version 3.11.0 -- Tue Jun 24 03:23:59 EST 2014

15:10:29 [SUCCESS] BNE-R01-IM4248 10.10.0.2:22
OpenGear/IM42xx Version 3.11.0 -- Tue Jun 24 03:23:59 EST 2014

15:10:29 [SUCCESS] BNE-R01-IM4248 10.10.0.3:22
OpenGear/IM42xx Version 3.11.0 -- Tue Jun 24 03:23:59 EST 2014
```

There are a number of components to this output:

1. The first line displays the *run ID*. This is the unique ID for this command including a timestamp of when the

- command was run, used to locate historical logs (discussed below).
- Each node command result has a result *header* line. This contains the time the command completed, if the command succeeded or failed, the node name, and the node address.
 - The *output* (stdout) of the command being run is listed for each node on which the command was run. If there is no output, only the header line is listed.

There are a few ways to modify the output of the command, useful for batch operation or noisy commands. To hide the command output results, use the `--quiet` argument and only the headers will be shown. To suppress headers and display command output only, use the `--batch` argument. Combine both arguments to hide all output.

Output Format: Logs

Information about each run is logged to the filesystem, by default. Filesystem logging can be disabled by using the `--disable-fslog` argument. The logs are stored in `/var/nvlog/node-command/`, and are indexed using the run ID of each command (as detailed in the Output Format: CLI section).

A new directory is created for each run, and contains 3 things:

- A *targets.txt* file, listing the addresses on which the command was run.
- a *stdout* directory, containing the output to stdout for the command printed by each console server.
- a *stderr* directory, containing the output to stderr for the command printed by each console server.

By default, a history of the last 30 commands are kept logged to the file system before being removed.

Syslog Information

In addition to the output and file logging, the running of commands is also recorded in syslog (**Status: Syslog** in the UI or `/var/log/messages` from the CLI).

```
<11>Mar 12 15:10:29 node-command[29534]: User 'root' ran command 'cat /etc/version'
on node '10.10.0.1:22'
<11>Mar 12 15:10:29 node-command[29534]: User 'root' ran command 'cat /etc/version'
on node '10.10.0.2:22'
<11>Mar 12 15:10:29 node-command[29534]: User 'root' ran command 'cat /etc/version'
on node '10.10.0.3:22'
```

4.3.2 node-upgrade Bulk Firmware Upgrade

The node-upgrade tool is used to upgrade the firmware of managed console servers, allowing administrators to easily perform firmware upgrades on all or on a range of their console server deployment.

Similar to the other node- commands, node-upgrade uses the SSH tunnel to transfer the firmware image image from Lighthouse CMS and initiate the upgrade. So unlike the Command Console Servers upgrade command, access to an external HTTP server is not required, making it ideal for firewalled environments.

Note The node-upgrade command requires the remote console server to have USB local storage available on the selected console server to store the firmware image prior to upgrade. Most Opengear products have internal flash storage that is suitable for this purpose, apart from the CM4100 console servers, SD4000 device servers and the ACM5000 Remote Site Managers without a -F designation.

Getting Started

The node-upgrade tool is based on the node-command tool, and many basic arguments such as displaying help and selecting console servers are the same. Please refer to the node-command Getting Started and Selecting Console Servers sections before continuing.

Staging Firmware on Lighthouse CMS

First, node-upgrade needs the firmware file to be available on the Lighthouse CMS's filesystem.

For temporary storage of the files, they can be placed at /tmp, otherwise, the non-volatile storage at /var/nvlog is a good place to store firmware images.

The appropriate firmware image can be copied on to the Lighthouse CMS's filesystem with using a program like WinScp from Windows PCs or scp CLI from OS X, Linux and other Unix-like systems, e.g.:

```
scp acm550x-3.11.1.flash root@lighthouse:/var/nvlog/
```

Upgrading Firmware

Run node-upgrade with the `--firmware-file` parameter, specifying the full path to the firmware file staged in the previous step, and the appropriate arguments selecting the console servers to upgrade.

e.g. to upgrade all ACM5500s to 3.11.1 that have not been upgraded already:

```
node-upgrade ---select-match 'Model=ACM55.*' --deselect-match 'Version=3.11.1' \  
  --firmware-file /var/nvlog/acm550x-3.11.1.flash
```

4.3.1 node-user Suite Bulk User Management

The node-user suite of tools are used to modify the user database on managed console servers, allowing administrators to easily add, remove and modify local users in bulk, on all or on a range of their console server deployment.

The individual command names are:

```
node-user-add  
node-user-del  
node-user-mod
```

Getting Started

The node-user tools are based on the node-command tool, and many basic arguments such as displaying help and selecting console servers are the same. Please refer to the node-command Getting Started and Selecting Console Servers sections before continuing.

Adding a User

Run `node-user-add --help` to display the arguments and syntax for adding a user:

```
usage: node-user-add [options] username  
username          Username to add  
-G --group-list  list      List of groups to give membership to for this user  
-C --port-list   list      List of ports numbers to give access for this user  
-T --description desc    User viewable description for this user  
-X --no-prompt  password  Set the users password  
-P --password                               Prompt for a password
```

e.g. to a user on all console servers with a username of *myadmin*, a description of *My Administrator* and membership of the admin group:

```
node-user-add --all --group-list admin --description "My Administrator" myadmin
```

Deleting a User

Run `node-user-del --help` to display the arguments and syntax for deleting a user:

```
usage: node-user-del [options] username [username ...]
       username [username ...]      List of users to delete
```

e.g. to delete the user *myadmin* and *myuser* from all console servers:

```
node-user-del --all myadmin myuser
```

Modifying a User

Run `node-user-mod --help` to display the arguments and syntax for modifying a user:

```
usage: node-user-mod [options] username
       username                Username to modify
  -G --group-list list        List of groups to add membership to for this user
  -C --port-list list        List of ports numbers to grant access for this user
  -T --description desc      User viewable description for this user
  -L --lock-user             Lock this user from accessing the device
  -U --unlock-user          Unlock this user from accessing the device
  -X --no-prompt password    Set the user's password
  -P --password              Prompt for a password
```

Lock and unlock temporarily disables and re-enables a user's ability to login to console server (establish sessions are not affected).

Port list is a list of serial ports a user account is explicitly permitted to access. Each port number can be preceded by a + or a - character. If a port number is preceded by a + the port is added to the user's explicit permissions list. If a port number is preceded by -, the port is removed from the user's explicit permissions list. Note that removed a port may not revoke access to a port, if the user has inherited permissions to access it by some other means (e.g. group permissions or admin group membership).

Similarly, the + and - syntax can be used when specifying the group list to add and remove group membership. If neither + nor - precedes a port or group, + is assumed.

e.g. to add *myuser* to the users group and grant permission to access serial port 1 on all console servers:

```
node-user-mod --all --port-list 1 --group-list users myuser
```

Passwords

For operations that require a password, such as `node-user-add` or `node-user-mod` with a `-P` or `-X` option, there are two ways that that password can be obtained. By default, when a password is required, it interactively prompts the administrator running the command for the password.

Alternatively, specify the password on the command line with the `-X` option, but be aware this means that the user's password will appear in plaintext in any `ps` process listings. The password is then encrypted before being sent across to the remote console server so that it does not appear in any logs in plaintext.

Synchronizing Console Servers

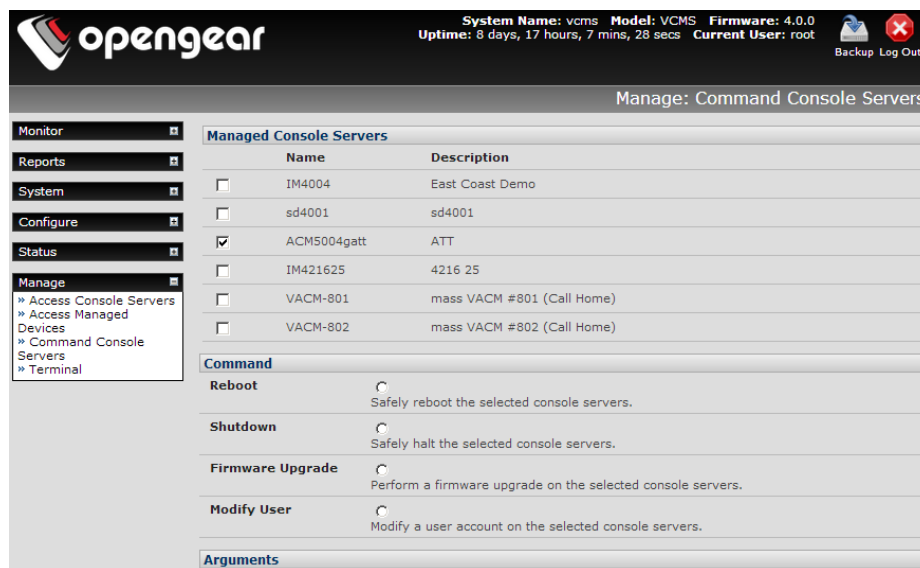
As users are added and deleted on the remote console servers, the user database on the Lighthouse CMS needs to be kept in synchronicity with the remote user databases. At the end of each `node-user-add`, `node-user-mod` and `node-user-del`, the administrator is prompted to resynchronize the affected remote console servers.

The synchronization is equivalent to the administrator had navigating to the **Configure: Managed Console Servers UI** and performing a Retrieve Managed Devices step. Alternatively, the behavior can be forced with either a -R (to always retrieve) or a -N (to not retrieve) option.

4.3.4 Command Console Servers UI

- Select **Manage: Command Console Servers** to display the list of *Managed Console Servers* that can be commanded by the current user. These are the *console servers* on which the current user has 'admin' group privileges

Note Only if the current user has 'admin' group privileges on a *console server*, are they deemed to be allowed to command that *console server*



- Check to select the **Managed Console Server(s)** to command
- Select the **Command** to schedule:
 - **Reboot:** Soft reboot the selected console servers
 - **Shutdown:** Halt the selected console servers. After being shut down, manual intervention in the form of a physical power cycle is required before the console server becomes available again
 - **Firmware Upgrade:** Perform a firmware upgrade, loading firmware from a given http:// URL, e.g. <http://www.opengear.com/firmware/acm500x-x.y.z.flash>

Command	
Reboot	<input type="radio"/> Safely reboot the selected console servers.
Shutdown	<input type="radio"/> Safely halt the selected console servers.
Firmware Upgrade	<input checked="" type="radio"/> Perform a firmware upgrade on the selected console servers.
Modify User	<input type="radio"/> Modify a user account on the selected console servers.
Arguments	
No command selected.	
URL	<input type="text"/> The URL from which to load the firmware file, must begin with http://.



It is important that the correct firmware file (i.e. one which matches the particular device type of the Managed Console Server) is uploaded. This is especially important when uploading firmware on multiple devices. Failure to do so could result in the need to net boot the device to recover which in turn requires physically visiting the device

- **Modify User:** Specify the *Username* to modify, the *Modification* to apply. Currently supported *Modifications* are *Lock Account* and *Unlock Account* where *Lock Account* prevents a user from logging in to the *console server* itself, or accessing *Managed Devices* using *SDT Connector* via the console server. Use *Unlock Account* to undo this modification.

Command	
Reboot	<input type="radio"/> Safely reboot the selected console servers.
Shutdown	<input type="radio"/> Safely halt the selected console servers.
Firmware Upgrade	<input type="radio"/> Perform a firmware upgrade on the selected console servers.
Modify User	<input checked="" type="radio"/> Modify a user account on the selected console servers.
Arguments	
Modification	Lock Account The modification to apply to the selected user account.
Username	<input type="text"/> Username of the account to modify.

- Click **Schedule Command**. The results of the schedule commands are displayed under **Monitor: Services** in the *Status Information* of the *Managed Console Server's Console server command*

System Name: vcms **Model:** VCMS **Firmware:** 4.0.0
Uptime: 8 days, 17 hours, 16 mins, 4 secs **Current User:** root

Monitor: Services

Current Network Status
 Last Updated: Sun Feb 3 01:47:38 EST 2013
 Updated every 90 seconds
 Nagios® 3.1.2 - www.nagios.org
 Logged in as root

Host Status Totals

Up	Down	Unreachable	Pending
38	7	0	0

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
78	1	2	40	41

All Problems 7 **All Types** 45 **All Problems** 43 **All Types** 162

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ACM5504-5-w	Auto-Response: name	PENDING	N/A	8d 10h 45m 25s+	1/1	Service is not scheduled to be checked...
	Command line shell	OK	2013-02-03 01:40:11	8d 16h 3m 31s	1/1	TCP OK - 0.011 second response time on port 23
	Console server command	PENDING	N/A	8d 10h 45m 25s+	1/1	Service is not scheduled to be checked...
	Firmware version	OK	2013-02-03 00:45:02	8d 16h 7m 31s	1/1	OpenGear/ACM51 Version 3.6.1b0 Tue Jan 8 16:58: EST 2013
	Management Console	OK	2013-02-02 23:35:11	8d 15h 59m 31s	1/1	TCP OK - 0.011 second response time on port 80
ACM5504-5-w - APC750	UPS APC750 Log	OK	2013-02-03 01:40:11	8d 16h 3m 31s	1/1	Status: OL Load: 12
	UPS APC750 Power	OK	2013-02-02 23:35:11	8d 15h 59m 31s	1/1	Input Voltage: 12 Battery Charge:

4.4 Manage Terminal

There are two methods available for accessing the CMS command line directly from a web browser:

System Name: vcms **Model:** VCMS **Firmware:** 4.0.0
Uptime: 8 days, 17 hours, 57 mins, 13 secs **Current User:** root

Manage: Terminal

Terminal
 The Web Terminal system service is not enabled. [Click here](#) to enable.

SDTConnector
Note: To access the OpenGear 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.

[Connect via SDTConnector](#)

- Select **Manage: Terminal**
- **Click here** in **Terminal** to activate the Web Terminal service. This uses AJAX to enable the web browser to connect to the *Lighthouse* appliance using HTTP or HTTPS, as a terminal - without the need for additional client installation on the user's PC
- The **Connect via 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

MONITORING WITH NAGIOS

5.1 Monitor

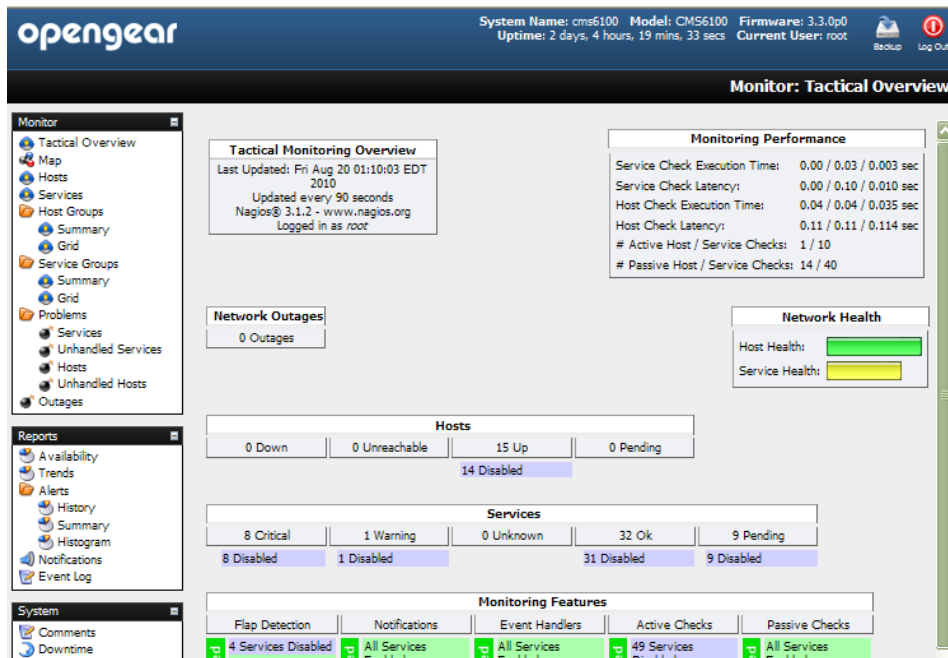
This section covers the Monitor menu options. The *CMS* monitoring software in your *Lighthouse* appliance is built on Nagios Core. All status screens under Monitor automatically refresh every 30 seconds, so there is no need to reload them (and this refresh time can be changed to even lower values in the *CMS* Nagios configuration files).

5.1.1 Tactical Overview

This screen gives you an overview of the current status of the monitored services and hosts.

Look at the *Hosts* and you see that you are currently monitoring 15 hosts (i.e. these will be the *Managed Console Servers* and their attached *Managed Devices*) and they are all *Up*. In the *Services* line you see that many of the services you are monitoring are disabled and report various levels of warning/critical status.

As a summary the *Network Health - Host* health bar on right is filled completely with green, indicating all configured hosts are OK while the *Service* health bar is filled with yellow.



Most fields on this page are links to more specific views e.g. if you wanted to see more details about your monitored services you can either click on the *8 Critical* field within the *Services* table (as shown below) or select *Problems: Services* from the Monitor menu:

System Name: cms6100 **Model:** CMS6100 **Firmware:** 3.3.0p0
Uptime: 2 days, 4 hours, 50 mins, 48 secs **Current User:** root

Monitor: Tactical Overview

Current Network Status
 Last Updated: Fri Aug 20 01:30:44 EDT 2010
 Updated every 90 seconds
 Nagios® 3.1.2 - www.nagios.org
 Logged in as root

Host Status Totals

Up	Down	Unreachable	Pending
15	0	0	0
All Problems		All Types	
0		15	

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
32	1	0	8	9
All Problems		All Types		
9		50		

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ACMS004 - CVS	Permitted Service - 1494/tcp - ica	CRITICAL	2010-08-20 01:28:53	2d 1h 30m 32s	1/1	Connection refused
	Permitted Service - 23/tcp - telnet	CRITICAL	2010-08-20 01:28:53	2d 1h 30m 32s	1/1	Connection refused
	Permitted Service - 3389/tcp - rdp	CRITICAL	2010-08-20 01:27:53	2d 1h 28m 32s	1/1	Connection refused
CM4001 - CVS	Permitted Service - 5900/tcp - vnc	CRITICAL	2010-08-20 01:29:53	0d 21h 33m 17s	1/1	Connection refused
	Permitted Service - 1494/tcp - ica	CRITICAL	2010-08-20 01:29:33	2d 1h 27m 19s	1/1	Connection refused
	Permitted Service - 23/tcp - telnet	CRITICAL	2010-08-20 01:29:33	2d 1h 27m 19s	1/1	Connection refused
	Permitted Service - 3389/tcp - rdp	CRITICAL	2010-08-20 01:28:33	2d 1h 25m 21s	1/1	Connection refused
	Permitted Service - 5900/tcp - vnc	CRITICAL	2010-08-20 01:30:32	0d 21h 32m 50s	1/1	Connection refused

5.1.2 Hosts

This screen shows the details of all the monitored hosts (i.e. all the *Managed Console Servers* in your distributed network and all the *Managed Devices* that are attached to them at the local and remote sites). You will see all configured hosts and have the choice to select one to get more information about it.

System Name: cms6100 **Model:** CMS6100 **Firmware:** 3.3.0p0
Uptime: 2 days, 4 hours, 45 mins, 42 secs **Current User:** root

Monitor: Hosts

Current Network Status
 Last Updated: Fri Aug 20 01:27:08 EDT 2010
 Updated every 90 seconds
 Nagios® 3.1.2 - www.nagios.org
 Logged in as root

Host Status Totals

Up	Down	Unreachable	Pending
15	0	0	0
All Problems		All Types	
0		15	

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
32	1	0	8	9
All Problems		All Types		
9		50		

Host Status Details For All Host Groups

Host	Status	Last Check	Duration	Status Information
ACMS004	UP	2010-08-20 01:26:54	2d 1h 26m 56s	OK
ACMS004 - CVS	UP	2010-08-20 01:26:54	2d 1h 26m 56s	OK
ACMS004 - Eaton	UP	2010-08-20 01:26:54	2d 1h 26m 56s	OK
ACMS004 - baytech	UP	2010-08-20 01:26:54	2d 1h 24m 56s	OK
CM4001	UP	2010-08-20 01:26:32	2d 1h 23m 43s	OK
CM4001 - Baytech	UP	2010-08-20 01:26:32	2d 1h 21m 45s	OK
CM4001 - CVS	UP	2010-08-20 01:26:32	2d 1h 23m 43s	OK
CM4001 - Level_2_Rm44_Port_1_EMD	UP	2010-08-20 01:26:32	2d 1h 23m 43s	OK
CM4116	UP	2010-08-20 01:26:51	2d 1h 22m 15s	OK
IM4004	UP	2010-08-20 01:26:31	2d 1h 25m 37s	OK

As we saw in the Tactical screen, here are the fifteen hosts we monitor right now. You can see basic information about each host on this page:

- **Host** shows all the hosts which are configured (If this field is marked red, the host itself is down, if it's just grey the server is up and reachable with ping, and if green then the host is OK)
- **Status** shows the current status of the hosts (OK = green, Warning = yellow, Critical = red, Unknown = orange)

- **Last Check** shows date and time when it has been checked the last time
- **Duration** shows for how long the service in this status
- **Status Information** is the output from the check program itself

And if you want to know more about a single host you select it by its name and you are redirected to a more detailed page about it.

5.1.3 Services

Similar to the *Hosts* view, *Services* shows the details of all the monitored screens. Again you see all configured services and have the choice to select one to get more information about it.

The screenshot displays the Nagios Services monitoring interface. At the top, the system name is 'cmr6100' with model 'CMS6100' and firmware '3.3.0p0'. The uptime is '2 days, 4 hours, 34 mins, 45 secs' and the current user is 'root'. The page title is 'Monitor: Services'.

Summary statistics are provided in three boxes:

- Current Network Status:** Last Updated: Fri Aug 20 01:14:41 EDT 2010. Updated every 90 seconds. Nagios@ 3.1.2 - www.nagios.org. Logged in as root.
- Host Status Totals:**

Up	Down	Unreachable	Pending
15	0	0	0
All Problems		All Types	
0		15	
- Service Status Totals:**

Ok	Warning	Unknown	Critical	Pending
32	1	0	8	9
All Problems		All Types		
9		50		

The main table, 'Service Status Details For All Hosts', shows details for host ACMS5004:

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ACMS5004	Alert 1 - test	WARNING	2010-08-18 00:28:33	2d 0h 46m 32s	1/1	1 alert current User root performed a logout on port02 (/dev/port02) TCP OK - 0.025 second response time on port 23
	Command line shell	OK	2010-08-20 01:13:53	2d 1h 14m 28s	1/1	Connect
	Console server command	PENDING	N/A	2d 1h 8m 13s+	1/1	Service is not scheduled to be checked...
	Firmware version	OK	2010-08-20 01:12:53	2d 1h 12m 28s	1/1	OpenGear/ACMS500v: Version 3.3.0p0 - Wed Aug 18 07:52:39 EST 2010 TCP OK - 0.017 second response time on port 80
	Management Console	OK	2010-08-20 01:11:53	2d 1h 13m 28s	1/1	Connect

The screen fields are also similar to *Hosts* (and all being well, the screen will all be grey and green - indicating there are no service problems). Only one additional field is displayed:

- **Attempt** shows how many attempts were needed for the check

5.1.4 Problems

These screens show the current problems with the hosts and services being monitored e.g. whenever a service reports a failure (like a connection alerts as shown below) you will get the information on this page.

The screenshot displays the OpenGear Monitor: Services dashboard. At the top, system information includes 'System Name: cms6100', 'Model: CMS6100', 'Firmware: 3.3.0p0', and 'Uptime: 2 days, 5 hours, 7 mins, 59 secs'. The current user is 'root'. The dashboard is divided into several sections: a left-hand navigation menu with options like 'Tactical Overview', 'Map', 'Hosts', 'Services', and 'Problems'; a 'Current Network Status' box with update frequency and Nagios version; 'Host Status Totals' and 'Service Status Totals' summary tables; a 'Display Filters' section; and a main table titled 'Service Status Details For All Hosts'. This table has columns for Host, Service, Status, Last Check, Duration, Attempt, and Status Information. One entry shows a 'WARNING' status for 'Alert 1 - test' on host 'ACM5004'.

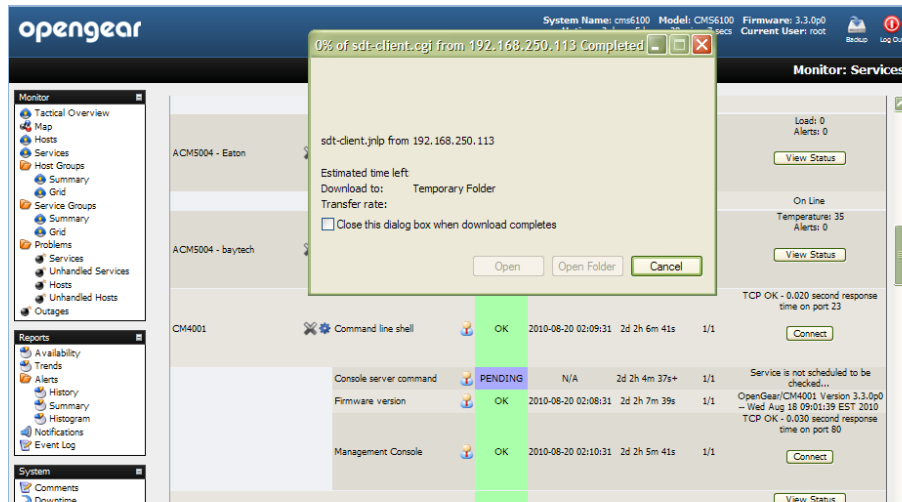
The browser refreshes every 30 seconds so you get the current list of failed services. Also CMS checks the hosts and services at regular (programmable) intervals. So if an error was reported, but on the next check reports that everything is okay for that service, the status will be updated. For example, CMS connects to each of the configured *Managed Console Servers* and their attached *Managed Devices* using all the services it was told are configured. If a service (like HTTP or SSH access) is momentarily disabled on a particular *Managed Device*, then the *Problems: Current Status: Services* will report a *Connection Refused* error, and this report will be removed when the service has been re-enabled.

5.1.5 Connecting with SDT Connector

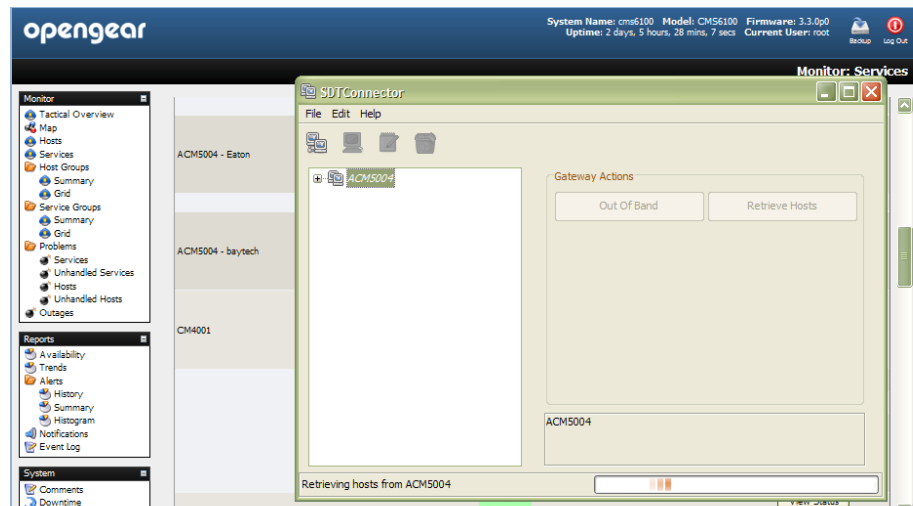
Many of the hosts displayed on the Monitor: Services screen have a **Connect**, **Manage Power**, **View Status** or **View Logs** button in the *Status Information* field as shown below.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ACM5004 - Eaton	UPS Eaton Log	OK	2010-08-20 02:06:52	2d 2h 5m 50s	1/1	View Status
	UPS Eaton Power	OK	2010-08-20 02:07:51	2d 2h 7m 50s	1/1	On Line
ACM5004 - baytech	RPC baytech	OK	2010-08-20 02:06:52	2d 2h 5m 50s	1/1	View Status
	Command line shell	OK	2010-08-20 02:06:31	2d 2h 3m 39s	1/1	Connect
CM4001 - Baytech	Console server command	PENDING	N/A	2d 2h 1m 35s+	1/1	Service is not scheduled to be checked...
	Firmware version	OK	2010-08-20 02:05:32	2d 2h 4m 37s	1/1	OpenGear/CM4001 Version 3.3.0p0 -- Wed Aug 18 09:01:39 EST 2010 TCP OK - 0.080 second response time on port 80
	Management Console	OK	2010-08-20 02:07:31	2d 2h 2m 39s	1/1	Connect
CM4001 - Baytech	RPC Baytech	OK	2010-08-20 02:07:31	2d 2h 2m 39s	1/1	View Status

- Click on this button and you will be connected to the relevant screen on that *Managed Device* or *Managed Console Server*
 - Your browser will download a configured *SDT Connector* Java application from the CMS and it will run on your computer. This *SDT Connector* is preconfigured with the gateway details (that being the *Managed Console Server*) and the host details (which will be one of the *Managed Devices* attached to the *Managed Console Server*, or the *Managed Console Server* itself)

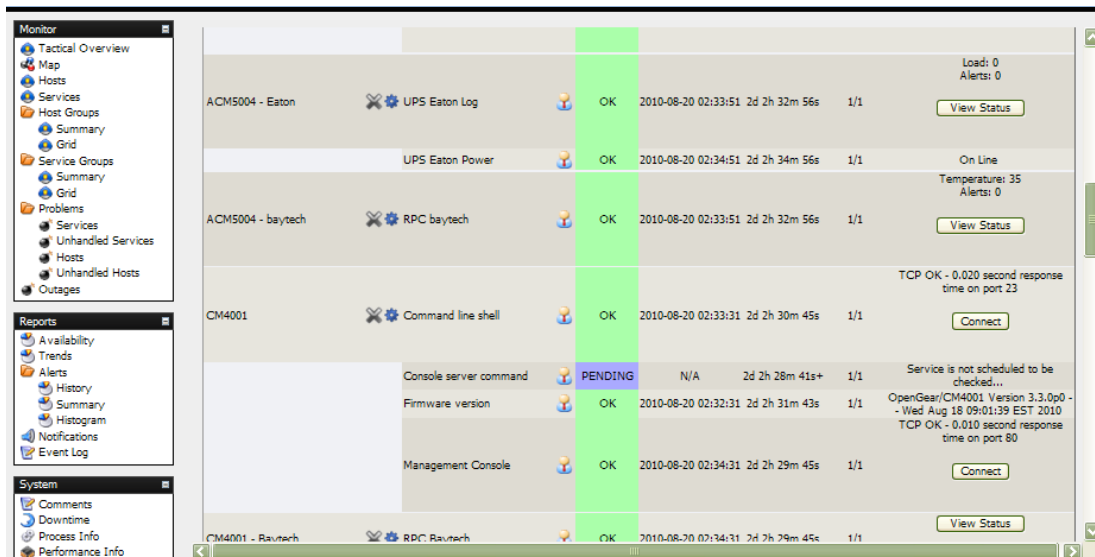


- *SDT Connector* will then log you into the SSH server embedded in the *Managed Console Server*, using the credentials of the user currently logged in to the *CMS*. Then, if appropriate, it will SSH tunnel connect you through to the target *Managed Device*

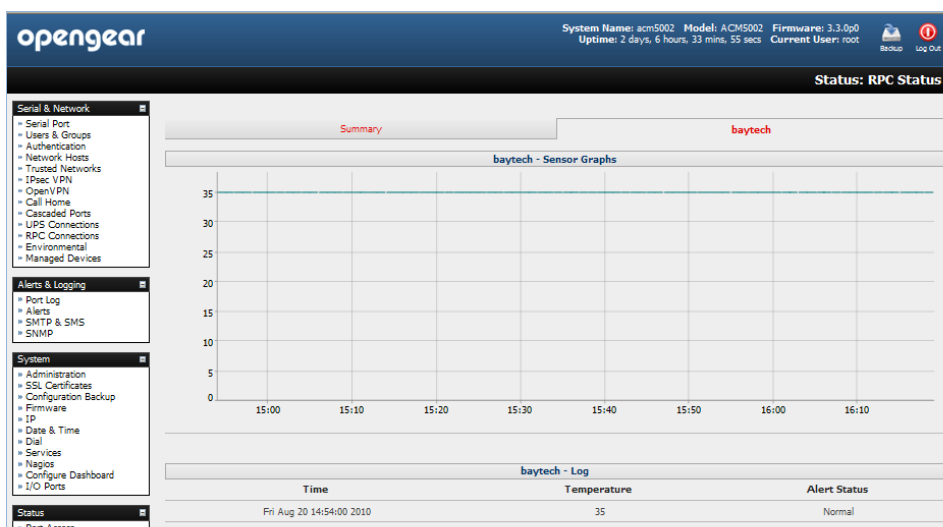


- Lastly *SDT Connector* will automatically load and run the appropriate application (*service*) on your computer that is needed to connect to the appropriate *Managed Device* or *Managed Console Server* screen.

This *service* could be a text-based console tool (such as SSH, telnet, SoL) or a browser/graphical/network tools (such as VNC, RDP, HTTPS, HTTP, X11, VMware, DRAC, iLO).



For example, if you clicked on the **View Status** button of the Monitor: Services screen, shown above, to get an update on the status of the BayTech RPC that is managed by a remote *Managed Console Server* named acm5002), the *SDT Connector* would launch and connect you the acm5002 *Managed Console Server*, and be presented with the *RPC: Status* display for the BayTech power device (shown below)

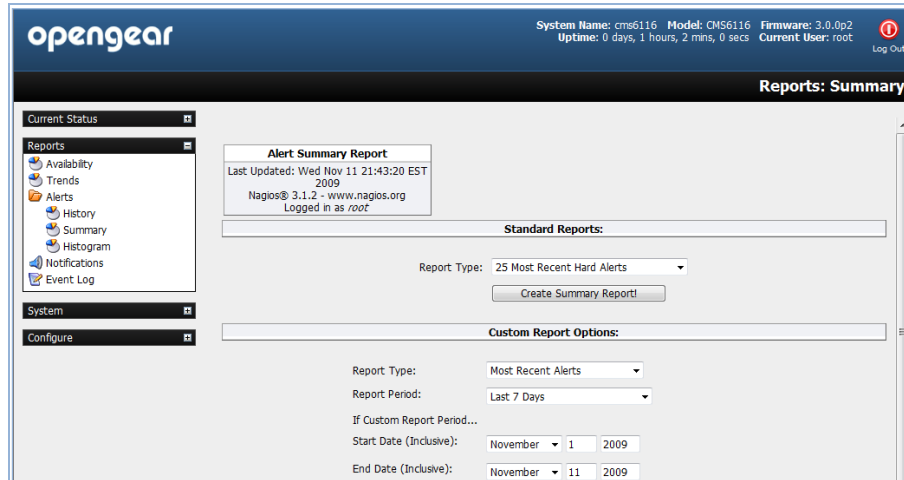


- So this connection is fully *point 'n click*

Note The location of the application which needs to be loaded and the appropriate commands to invoke it (e.g. which browser or SSH client software service will run) will vary from computer to computer. So you may need to configure the *SDT Connector* Java application with this information as detailed in Chapter 5. Alternatively, if you have a permanent *SDT Connector* client already installed on your computer, then when your browser downloads the preconfigured *SDT Connector* Java application it will, by default, use the *service* configurations already set up on your installed client.

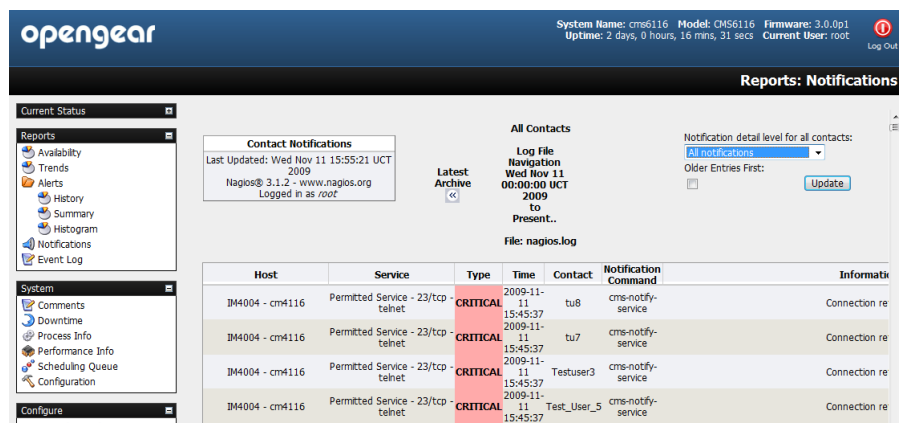
5.2 Reports and system

The CMS provides all the standard Nagios customizable reports and logs:



5.2.1 Notifications

All OpenGear *console servers* can be configured to send email and SMS alert notifications in event of an alert trigger event (pattern match on serial port, elevated temperature, door open etc). However, the Nagios features in CMS allow more sophisticated notification.



Host	Service	Type	Time	Contact	Notification Command	Information
IM4004 - cm4116	Permitted Service - 23/tcp-telnet	CRITICAL	2009-11-11 15:45:37	tu8	cms-notify-service	Connection re
IM4004 - cm4116	Permitted Service - 23/tcp-telnet	CRITICAL	2009-11-11 15:45:37	tu7	cms-notify-service	Connection re
IM4004 - cm4116	Permitted Service - 23/tcp-telnet	CRITICAL	2009-11-11 15:45:37	Testuser3	cms-notify-service	Connection re
IM4004 - cm4116	Permitted Service - 23/tcp-telnet	CRITICAL	2009-11-11 15:45:37	Test_User_5	cms-notify-service	Connection re

Basically, host and service *notifications* occur when a hard state change occurs, or when a host or service remains in a non-OK state for a specified period of time specified (since the last notification was sent out). CMS also allows for escalation of these notifications. For details on configuring notifications and escalations refer to the next section.

5.3 Extended Nagios

At the core of CMS's monitoring is Nagios (<http://www.nagios.org>) - the leading open source host, service and network monitoring tool. Nagios lets you manage different types of services and hosts running on different operating systems like Linux, Windows, and Solaris. It's flexible in configuration and can be extended. It's configured within text files and managed with a web browser.

When you do a basic *CMS* installation, you get a set of Nagios check programs which are automatically configured to let you start monitoring all the hosts and services on your *Managed Console Servers* and all their *Managed Devices*.

However, you can also extend the Nagios configuration to your special needs:

- You can add more check programs (refer to <http://www.nagiosexchange.org> where other developers have available their check programs for download)
- You can write your own in the supported programming languages (Bash, Perl)
- You can even have these new checks (NRPE and NCSA) running on your remote *Managed Console Servers* (to take load off the *CMS* and reduce network traffic)
- If you want, you can setup notifications with elevations
- You can extend the graphical web views of your managed hosts using NagVis

5.3.1 Adding custom checks + scripting/config set up

To submit additional check results to the *CMS*, make an NSCA connection to the loopback interface using *send_nsca* on the *Managed Console Server*:

```
send_nsca -H 127.0.0.1 -c /etc/config/node-send_nsca.cfg
```

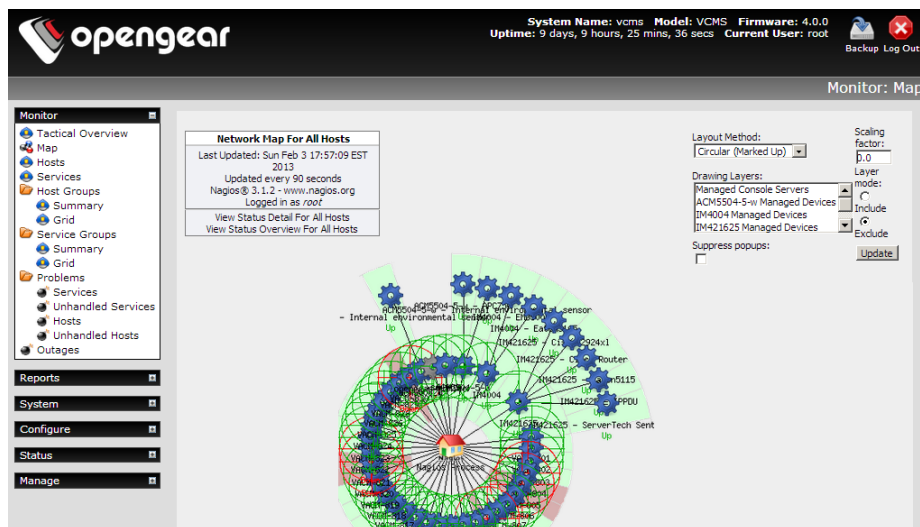
This port is securely tunneled back to the *CMS* NSCA server e.g. on the *Managed Console Server*, run:

```
printf "My Managed Host\tService Description\t0\tOK\n" | send_nsca -H  
127.0.0.1 -c /etc/config/node-send_nsca.cfg
```

The Nagios server on the *CMS* must have a service configured to receive the check result. Place custom Nagios configuration files in */etc/config/nagios/user/* on the *CMS*, then verify and (if successful) reload Nagios configuration with:

```
nagios -v /etc/config/nagios/nagios.cfg && pkill -HUP nagios
```

5.3.2 Introducing NagVis



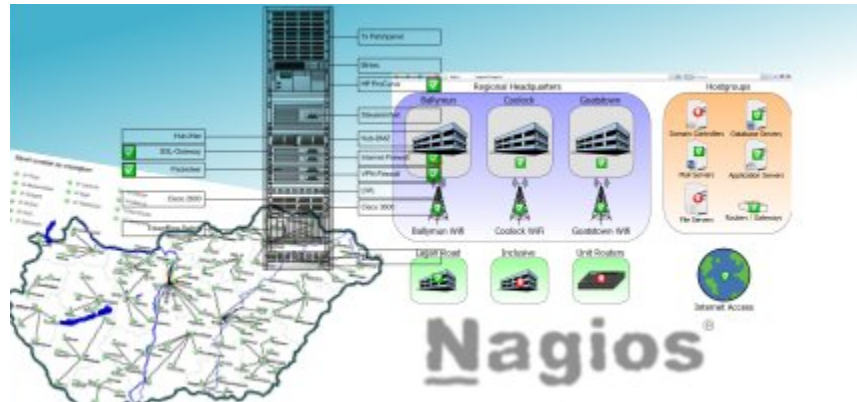
The standard **Monitor: Map** display in Nagios presents a basic image of the monitored host and service states. However, the NagVis1 add-on gives you a powerful flexible visualization tool for customizing the status display against any background image you choose.

NagVis can display different icons, depending on the state of the object (red for the CRITICAL state, yellow for WARNING, green for OK, and a question mark on a gray background for UNKNOWN). If an acknowledgment was set, this is indicated by a green button with a picture of a worker on it.

There are different icons for hosts and services. In the default template, host icons are rectangular and service icons are round. A finished NagVis *map* might present using a geographical map, or a photo of the server room as a background. In addition to hosts and services, host and service groups can also be integrated into a NagVis display, as well as additional maps. Thus a geographical overview map could be used for the start page, which has an icon for each location monitored that links to a detailed NagVis map specifically for that location.

If an icon contains several states, as is the case for host and service groups, for instance, NagVis displays the state with the highest priority. CRITICAL has a higher priority than WARNING, WARNING trumps UNKNOWN, UNKNOWN gets more attention than an acknowledgment, and OK has the lowest priority of all. If any host in a host group assumes the CRITICAL state, this is shown accordingly for the entire host group.

For hosts and host groups, NagVis offers you the choice of having only host states considered in determining the state that is displayed, or having the services dependent on these hosts are included as well (see page 394). In the latter case, a red stop light is displayed if even a single service of a host is in the critical state. For details on using NagVis refer www.nagvis.org



5.3.3 Notifications

All Opendgear *console servers* can be configured to send email and SMS alert notifications in event of an alert trigger event (e.g. a pattern match on serial port, elevated temperature or door open event). However the Nagios features in CMS allow more sophisticated notification.

Host	Service	Type	Time	Contact	Notification Command	Information
IM4004 - cm4116	Permitted Service - 23/tcp-telnet	CRITICAL	2009-11-11 15:45:37	tu8	cms-notify-service	Connection re
IM4004 - cm4116	Permitted Service - 23/tcp-telnet	CRITICAL	2009-11-11 15:45:37	tu7	cms-notify-service	Connection re
IM4004 - cm4116	Permitted Service - 23/tcp-telnet	CRITICAL	2009-11-11 15:45:37	Testuser3	cms-notify-service	Connection re
IM4004 - cm4116	Permitted Service - 23/tcp-telnet	CRITICAL	2009-11-11 15:45:37	Test_User_5	cms-notify-service	Connection re

With Nagios, host and service notifications occur when a hard state change occurs, or when a host or service remains in a hard non-OK state and the time specified (by the `<notification_interval>` option in the host or service definition) has passed since the last notification was sent out.

Each host and service definition has a `<contact_groups>` option that specifies what contact groups receive notifications for that particular host or service. Contact groups can contain one or more individual contacts.

When Nagios sends out a host or service notification, it will notify each contact that is a member of any contact groups specified in the `<contactgroups>` option of the service definition. Nagios realizes that a contact may be a member of more than one contact group, so it removes duplicate contact notifications before it does anything.

Just because there is a need to send out a host or service notification doesn't mean that any contacts are going to get notified. There are several filters that potential notifications must pass before they are deemed worthy enough to be sent out. Even then, specific contacts may not be notified if their notification filters do not allow for the notification to be sent to them. For example if the host or service is in a period of scheduled downtime. If it is in a scheduled downtime, no one gets notified.

The Nagios software can be configured to notify you of problems and recoveries pretty much anyway you want: pager, cell phone, email, instant message, audio alert, electric shocker, etc. How notifications are sent depend on the notification commands that are defined in your object definition files:

```
/etc/config/scripts/cms-notify-service
```

```
/etc/config/scripts/cms-notify-host
```

For more details refer http://nagios.sourceforge.net/docs/3_0/notifications.html

5.3.4 Notification Elevation

The Nagios software in *CMS* also supports optional escalation of contact notifications for hosts and services. Escalation of host and service notifications is accomplished by defining host escalations and service escalations in your object configuration file(s).

Notifications are escalated *if and only if* one or more escalation definitions match the current notification that is being sent out. If a host or service notification *does not* have any valid escalation definitions that apply to it, the contact group(s) specified in either the host group or service definition will be used for the notification.

Users can define service and host escalations in `/etc/config/nagios/user directory`

For more details refer http://nagios.sourceforge.net/docs/3_0/escalations.html

5.3.5 An example showing you how to add new check programs

This example adds a simple bash script that checks if the file `/tmp/nagios.chk` is available. If it is there and it's executable the service goes to critical, if it is there and not executable it's going to warning and if it doesn't exist the service is ok.

1. Create the executable check file

```
# vi /usr/local/nagios/libexec/check_file_exist.sh
```

Add the following to that file:

```
#!/bin/bash  
#  
# Check if a local file exist  
#  
while getopts F: VAR  
do
```

```

case "$VAR" in
F ) LOGFILE=$OPTARG ;;
* ) echo "wrong syntax: use $o -F <file to check>"
exit 3 ;;
esac
done

if test "$LOGFILE" = ""
then
echo "wrong syntax: use $O -F <file to check>"
# Nagios exit code 3 = status UNKNOWN = orange
exit 3
fi
if test -e "$LOGFILE"
then
if test -x "$LOGFILE"
then
echo "Critical $LOGFILE is executable !"
# Nagios exit code 2 = status CRITICAL = red
exit 2
else
echo "Warning $LOGFILE exists !"
# Nagios exit code 1 = status WARNING = yellow
exit 1
fi
else
echo "OK: $LOGFILE does not exist !"
# Nagios exit code 0 = status OK = green
exit 0
fi

```

Now set the file attributes:

```

# chown nagios.nagios /usr/local/nagios/libexec/check_file_exist.sh
# chmod +x /usr/local/nagios/libexec/check_file_exist.sh

```

Add the check program to the nagios configuration

Each new check command has to be defined once in the global Nagios configuration:

```

# vi /usr/local/nagios/etc/minimal.cfg

```

Add the following block at the end of the file:

```

define command{
command_name check_file_exist
command_line $USER1$/check_file_exist.sh -F /tmp/nagios.chk
}

```

Add a new service to the localhost. Each new service has to be defined once in the Nagios configuration and can be assigned to a single host, multiple hosts or even a host group. We assign it only to the localhost that is already defined in this base configuration:

```
# vi /usr/local/nagios/etc/minimal.cfg
```

Add the following block at the end of the file:

```
define service{
use generic-service
host_name localhost
service_description File check
is_volatile 0
check_period 24x7
max_check_attempts 4
normal_check_interval 5
retry_check_interval 1
contact_groups admins
notification_options w,u,c,r
notification_interval 960
notification_period 24x7
check_command check_file_exist
}
```

Verify Nagios configuration and restart it. After all changes of the config files you should check the Nagios configuration and you have to restart Nagios after that:

```
# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

The Total Warnings and Total Errors should be 0 if you have done everything correct.
So restart it with:

```
# /etc/init.d/nagios restart
```

Check if the new program is working. First take a look at the tactical screen and you should see that one service is in status pending. That means no check was done before for this service.

Wait a view minutes and it should disappear as pending and the number of OKs should increment from 5 to 6.

Now create the file and watch the tactical screen, the service detail screen or the service problems screen.

```
# touch /tmp/nagios.chk
```

As we set the *normal_check_interval* to 5 minutes in the service definition, you should get the warning message during that time. Now add the executable attribute and watch:

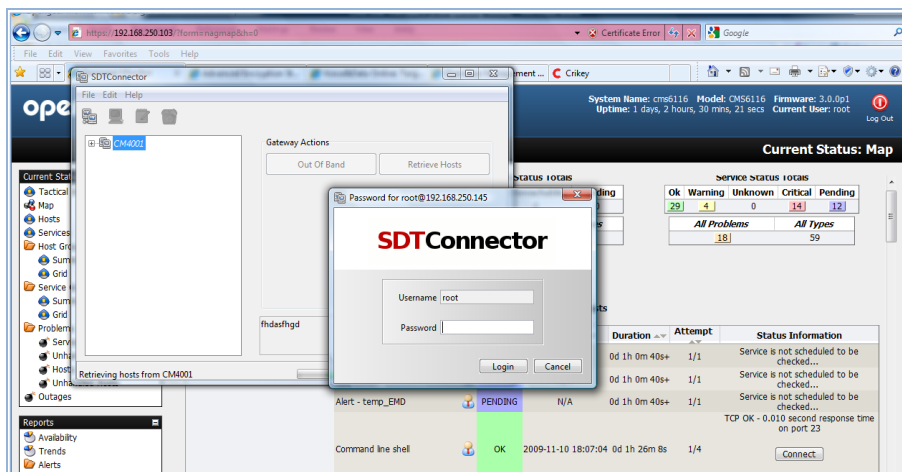
```
# chmod +x /tmp/nagios.chk
```

The status should change during the check interval to critical. When you delete the file the service should return to status ok.

ACCESSING WITH SDT CONNECTOR

This chapter describes using *SDT Connector* to securely communicate with *Managed Console Servers* and their attached *Managed Devices*. *SDT Connector* is a simple Java application that sets up secure SSH tunnels and then runs a local application.

As covered earlier, when you are browser connected to the *CMS* you can click on the **Connect** or **Manage Power** or **View Status** or **View Logs** button in the *Status Information* field of any monitored *Host* and browser will download a pre-configured *SDT Connector* Java application from the *CMS* and you will be connected to the *Host* (proxied via the *CMS*).



This pre-configured *SDT Connector* is preconfigured with the *gateway* details (that being the *Managed Console Server*) and the host details (which will be one of the *Managed Devices* attached to the *Managed Console Server*, or the *Managed Console Server* itself) and it will log you into the SSH server embedded in the *Managed Console Server* (you will need to enter a Username Password) and then automatically load and run the appropriate application (*service*) on your computer that is needed to connect to the appropriate *Managed Device* or *Managed Console Server* screen.

The *service* details (location of the application itself and commands to run) may need to be configured in the *SDT Connector* (refer Chapter 6.1). Alternatively if you have a permanent *SDT Connector* installed on your computer it will use the *service* configuration already set up there.

There are many advantages to having such a permanent installation and the balance of this chapter then covers such installation and configuration options:

- Configuring the *console server* for SSH tunneled access to network attached hosts and setting up permitted Services and user access (*Section 6.1*)
- Setting up the *SDT Connector* client with gateway, host, service and client application details and making connections between the Client PC and hosts connected to the *console server* (*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 SSH 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)

6.1 Configuring for SSH Tunneling to Hosts

To set up the *console server* for SSH tunneled access a network attached *host*:

- Add the new *host* and the *permitted services* using the **Serial & Network: Network Hosts** menu as detailed in *Network Hosts* (Chapter 4.4). Only these *permitted services* will be forwarded through by SSH 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 on a 48 port <i>console server</i>)
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 client installation and configuration

The *SDT Connector* client works with all Opengear *console servers*. Each of these remote *console servers* have an embedded OpenSSH based server which 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 to be run when access to a particular host has been established.

SDT Connector can connect to the *console server* using an alternate OoB access. It can also access the *console server* itself and 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 Opengear *console server* product (or a copy can be freely download from Opengear's website)
- Run the set-up program:



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 there is already a config file on the Windows PC then it will not be overwritten. To remove earlier config file run the *regedit* command and search for “*SDT Connector*” 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://www.java.com/getjava/>. It will install on Windows and on most Linux platforms. Solaris platforms are also supported however they must have Firefox installed. *SDT Connector* can run on any system with Java 1.4.2 and above installed, but it assumes the web browser is Firefox, and that *xterm -e telnet* opens a telnet window

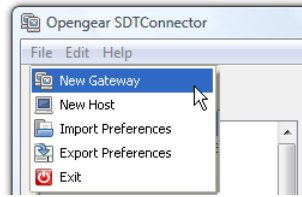
To operate *SDT Connector*, you first need to add new gateways to the client software by entering the access details for each *console server* (refer [Section 6.2.2](#)) then let the client auto-configure with all host and serial port connections from each *console server* (refer [Section 6.2.3](#)) then point-and-click to connect to the Hosts and serial devices (refer [Section 6.2.4](#))

Alternately you can manually add network connected hosts (refer [Section 6.2.5](#)) and manually configure new services to be used in accessing the *console server* and the hosts (refer [Section 6.2.6](#)) then manually configuring clients to run on the PC that will use the service to connect to the hosts and serial port devices (refer [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 [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

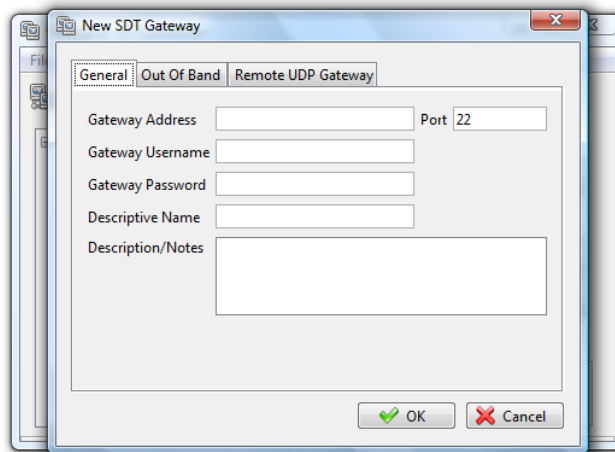


- Enter the IP or DNS **Address** of the *console server* and the SSH port that will be used (typically 22)

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 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

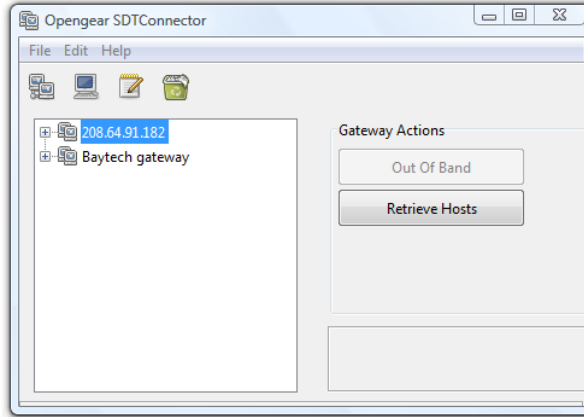


- Optionally, enter a **Descriptive Name** to display instead of the IP or DNS address, and any **Notes** or a **Description** of this gateway (such as its firmware version, site location or anything special about its network configuration).
- Click **OK** and an icon for the new gateway will now appear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

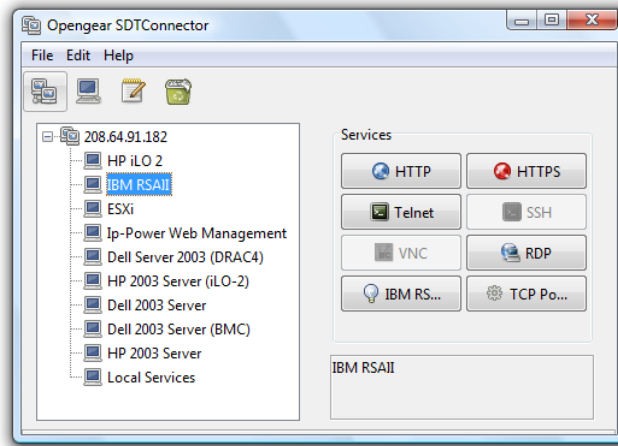
Note For an *SDT Connector* user to access a *console server* (and then access specific hosts or serial devices connected to that *console server*), that user must first be setup on the *console server*, and must be authorized to access the specific ports / hosts (refer Chapter 5) and only these *permitted services* will be forwarded through by SSH 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 which has been configured with those specific connected hosts and serial port devices the user has authority to access, and a specific set of the enabled services for each of these. This configuration can be auto-uploaded into the *SDT Connector* client:



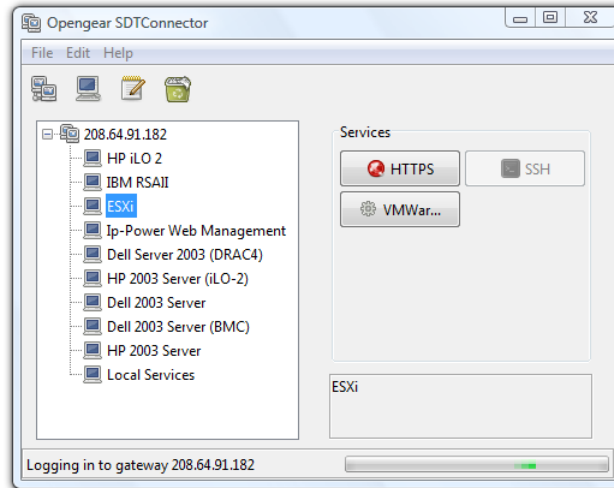
- Click on the new gateway icon and select **Retrieve Hosts**. This will:
 - configure access to network connected Hosts that the user is authorized to access and set up (for each of these Hosts) the services (e.g. HTTPS, IPMI2.0) and the related IP ports being redirected
 - configure access to the *console 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) however *SDT Connector* will not auto-configure the *root* (and it 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:




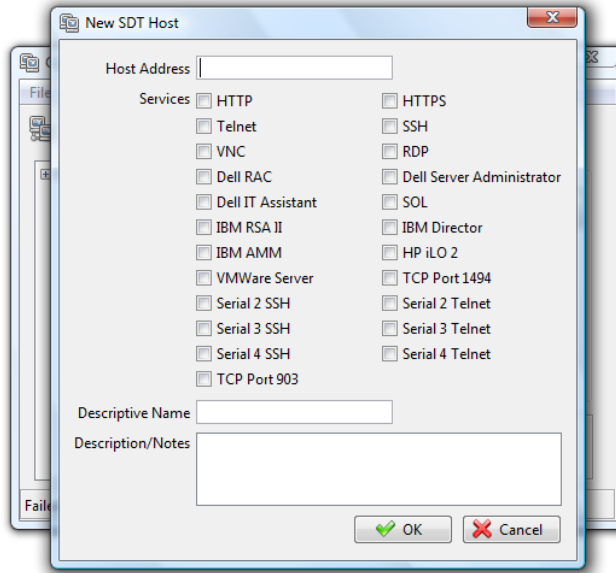
Note The *SDT Connector* client can be configured with unlimited number of Gateways. Each Gateway can be configured to port forward to an unlimited number of locally networked Hosts. Similarly there is no limit on the number of *SDT Connector* clients who can be configured to access the one Gateway. Nor are there 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 the one time on a particular Gateway. SD4002/4008 and CM4001/4008 devices support at least 10 simultaneous client tunnels; IM4216/4248 and CM4116/4148 each support at least 50 such concurrent connections. So for a site with a CM4116 gateway you can have, at any time up to 50 users securely controlling an unlimited number of network attached computers and appliances (servers, routers etc) at that site.

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 be 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**)

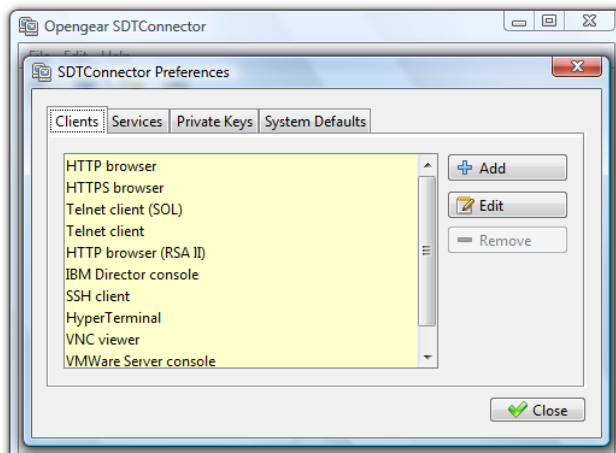


- Enter the IP or DNS **Host Address** of the host (if this is a DNS address, it must be resolvable by the gateway)
- Select which **Services** are to be used in accessing the new host. A range of service options are pre-configured in the default *SDT Connector* client (RDP, VNC, HTTP, HTTPS, Dell RAC, VMware etc). However if you wish to add new services the range then proceed to the next section (**Adding a new service**) then return here
- Optionally, enter a **Descriptive Name** for the host, to display instead of the IP or DNS address, and any **Notes** or a **Description** of this host (such as its operating system/release, or anything special about its configuration)
- Click **OK**

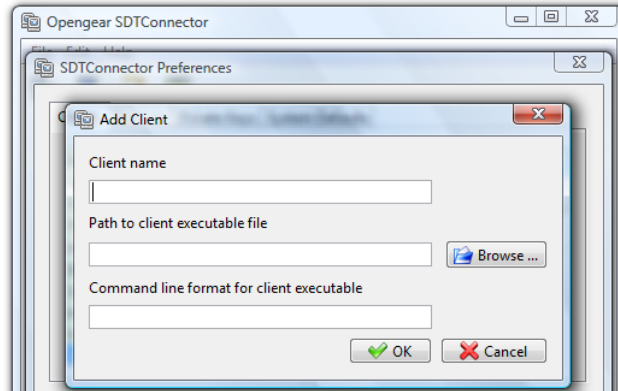
6.2.6 Manually adding new services to the new hosts

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 use to access the local endpoint of the redirection



- Select which **Client** application is associated with the new service. A range of client application options are pre-configured in the default *SDT Connector* (RDP client, VNC client, HTTP browser, HTTPS browser, Telnet client etc). However if you wish to add new client applications to this range then proceed to the next section (**Adding a new client**) then return here

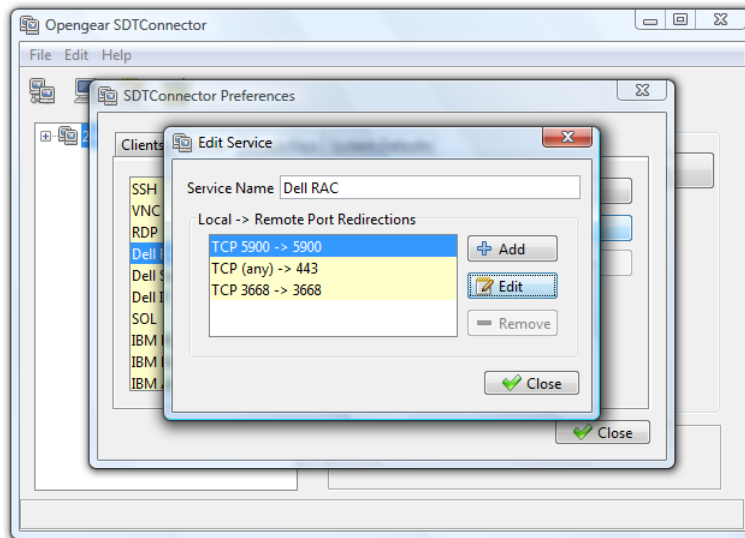


- Click **OK**, then **Close**

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.

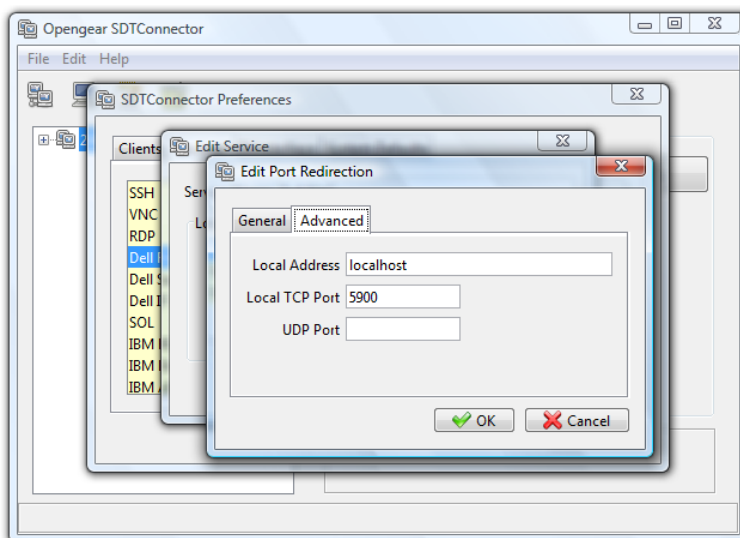
The second redirection is for the VNC service that the user may choose to later launch from the RAC web console. It is automatically loads in a Java client served through the web browser, so it does not need a local client associated with it.



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

You may also specify **Advanced** port redirection options:

- Enter the local address to bind to when creating the local endpoint of the redirection. It is not usually necessary to change this from "localhost".
- Enter a local TCP port to bind to when creating the local endpoint of the redirection. If this is left blank, a random port will be selected.



Note *SDT Connector* can also tunnel UDP services. *SDT Connector* tunnels the UDP traffic through the TCP SSH redirection, so in effect it is a tunnel within a tunnel.

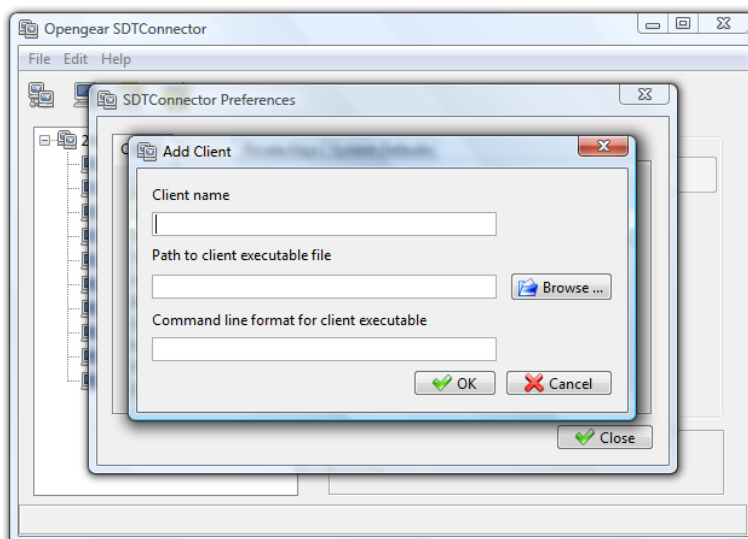
Enter the UDP port on which the service is running on the host. This will also be the local UDP port that *SDT Connector* binds as the local endpoint of the tunnel.

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)

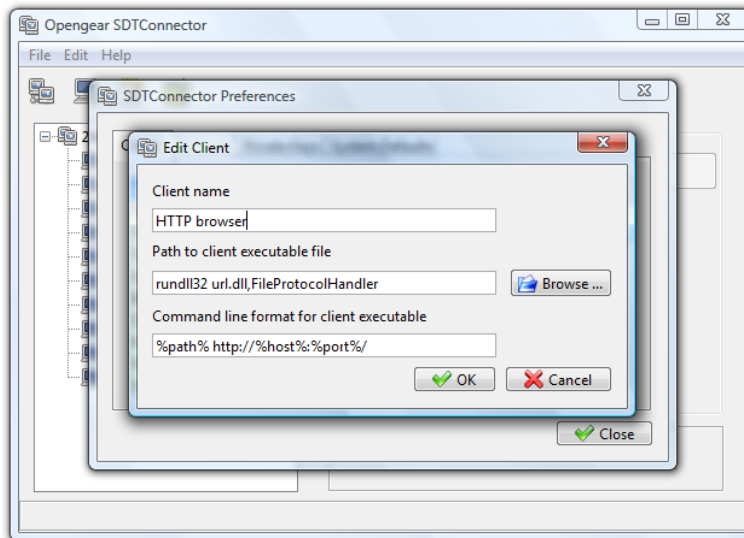
- Enter a **Command Line** associated with launching the client application. *SDT Connector* typically launches a client using command line arguments to point it at the local endpoint of the redirection. There are three special keywords for specifying the command line format. When launching the client,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

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

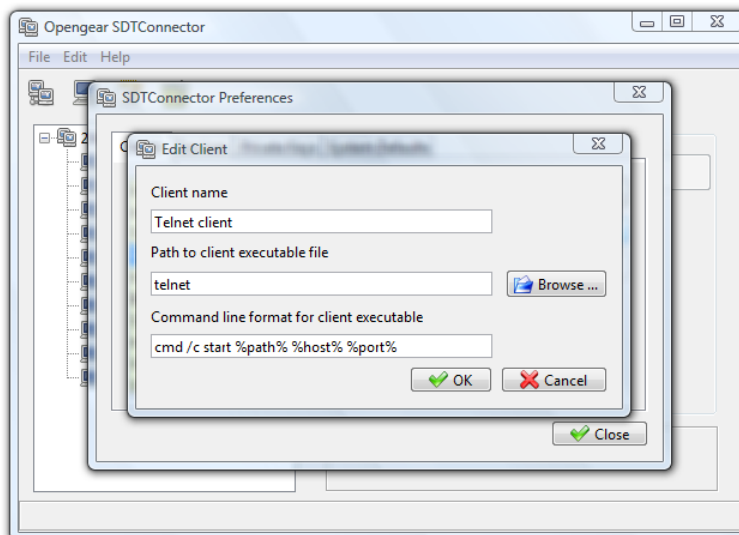
%host% is the local address to which the local endpoint of the redirection is bound, i.e. the Local Address field for the Service redirection Advanced options.

%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:



Also some clients are launched in a command line or terminal window. The Telnet client is an example of this so the "Path to client executable file" is *telnet* and the "Command line format for client executable" is `cmd /c start %path% %host% %port%` :



- Click **OK**

6.2.8 Dial in configuration

If the client PC is dialing into *Local/Console* port on the *console server* you will need to set up a dial-in PPP link:

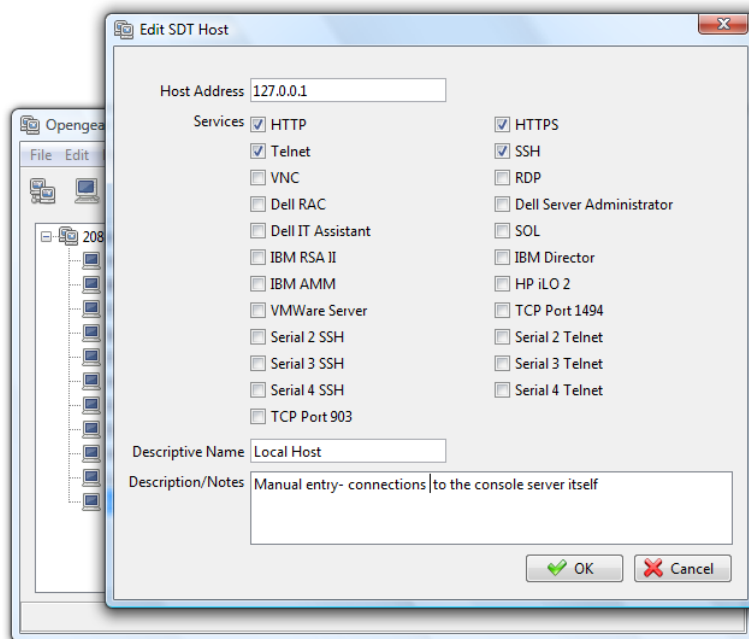
- Configure the *console server* for dial-in access (following the steps in the **Configuring for Dial-In PPP Access** section in *Chapter 5, Configuring Dial In Access*)
- Set up the PPP client software at the remote *User PC* (following the **Set up the remote Client** section in *Chapter 5*)

Once you have a dial-in PPP connection established, you then can set up the secure SSH tunnel from the remote Client PC to the *console server*.

6.3 SDT Connector to Management Console

SDT Connector can also be configured for browser access 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 PC. Assuming you have already set up the *console server* as a *Gateway* in your *SDT Connector* client (with *username/password* etc) select this newly added *Gateway* and click the Host icon to create a host. Alternatively, select **File -> New Host**
- Enter 127.0.0.1 as the **Host Address** and give some details in **Descriptive Name/Notes**. Click **OK**



- 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:

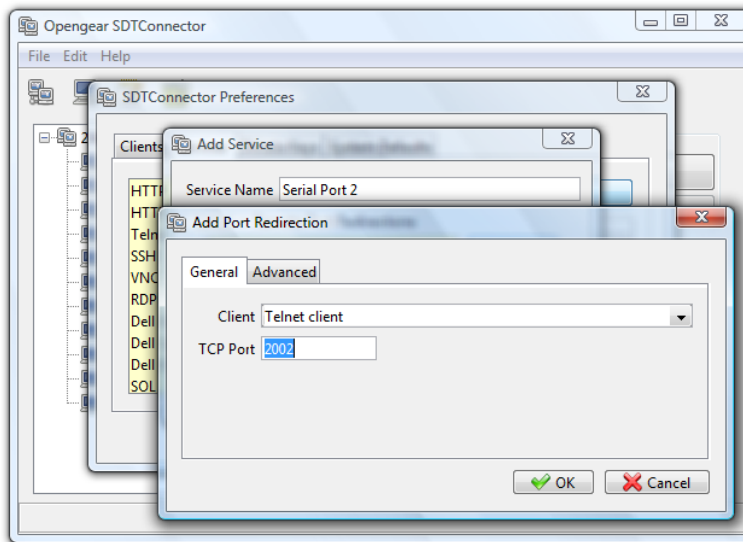
- Browse to the *console server* and select **Network Hosts** from **Serial & Network**, click **Add Host** and in the **IP Address/DNS Name** field enter 127.0.0.1 (this is the Opengear's network loopback address) and enter *Loopback* in **Description**

- 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) then 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 connect to serially attached devices

SDT Connector can also be used to access text consoles on devices that are attached to the *console server* 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 PC. 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



- 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** next 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

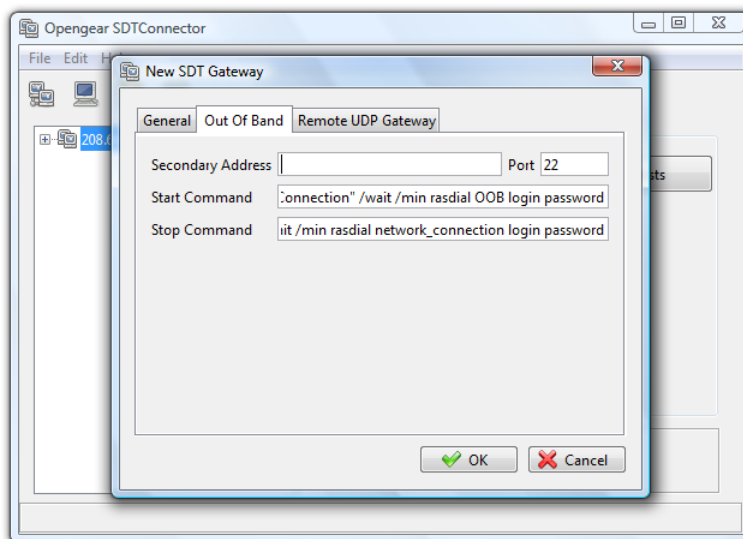
- Scroll down to *Console Server Setting* and select **Console server Mode**. Check **Telnet** (or SSH) and scroll to the bottom and click **Apply**
- Select **Network Hosts** from **Serial & Network** and click **Add Host**
- In the **IP Address/DNS Name** field enter *127.0.0.1* (this is the Opengear's 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 Using SDT Connector for out-of-band connection to the gateway

SDT Connector can also be set up to connect to the *console server* (gateway) out-of-band (OoB). OoB access uses an alternate path for connecting to the gateway to that used for regular data traffic. OoB access is useful for when the primary link into the gateway is unavailable or unreliable.

Typically a gateway'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 in this regard, by allowing you to provide your own scripts or commands for starting and stopping the OoB connection.



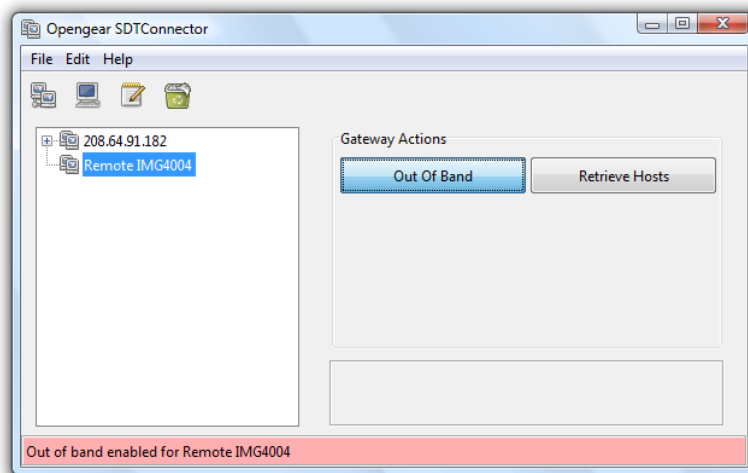
To configure *SDT Connector* for OoB access:

- When adding a new gateway or editing an existing gateway select the **Out Of Band** tab
- Enter the secondary, OoB IP address of the gateway (e.g. the IP address it is accessible using when dialed in directly). You also may 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**
 - 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`
Where *network_connection* 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.
 - To initiate a pre-configured dial-up connection under Linux, use the following Start Command:
`pon network_connection`
where *network_connection* is the name of the connection.
- Enter the command or path to a script to stop the OoB connection in **Stop Command**
 - 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`
where *network connection* is the name of the network connection as displayed in *Control Panel -> Network Connections*.
 - To stop a pre-configured dial-up connection under Linux, use the following Stop Command:
`poff network_connection`

To make the OoB connection using *SDT Connector*:

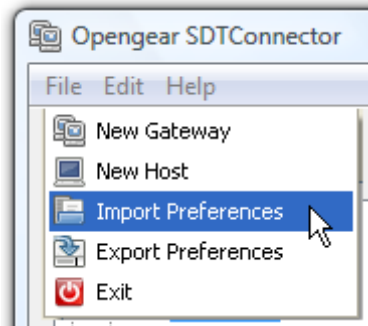
- Select the gateway and click Out Of Band. The status bar will change color to indicate this gateway is now being access using the OoB link rather than the primary link



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 you to enter your password. This is known as public key authentication.

To use public key authentication with *SDT Connector*, first you must add the public part of your SSH key pair to your SSH gateway:

- Ensure the SSH gateway allows public key authentication, this is typically the default behavior
- If you do not already have a public/private key pair for your client PC (the one running *SDT Connector*) generate them now using *ssh-keygen*, *PuTTYgen* or a similar tool. You may use RSA or DSA, however it is important that you leave the passphrase field blank:
 - PuTTYgen: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
 - OpenSSH: <http://www.openssh.org/>
 - OpenSSH (Windows): <http://sshhwindows.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 otherwise add to *.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.

Also 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 to securely access and manage remote Windows computers – to reconfigure applications and user profiles, upgrade the server's operating system, reboot the

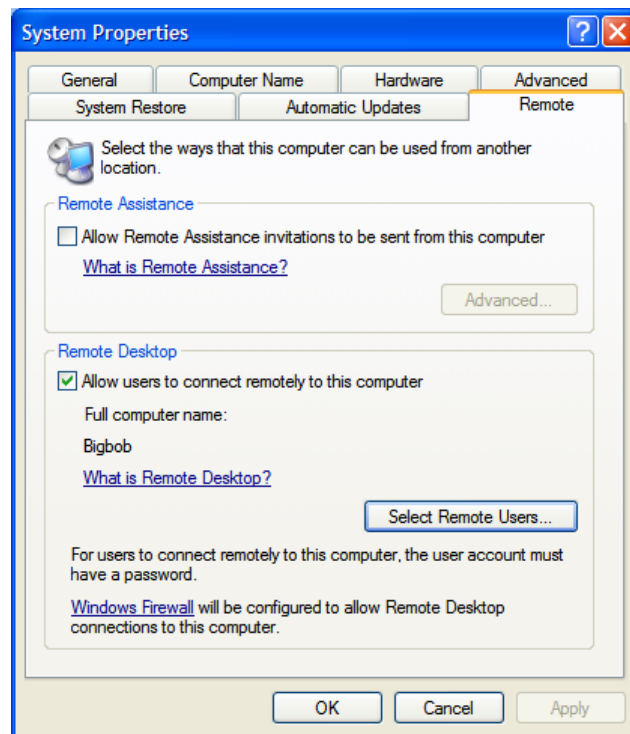
machine etc. Opengear's Secure Tunneling uses SSH tunneling, so this RDP traffic is securely transferred through an authenticated and encrypted tunnel.

SDT with RDP also allows remote *Users* to connect to Windows XP and later computers and to Windows 2000 Terminal Servers; and to have access to all of the applications, files, and network resources (with full graphical interface just as though they were in front of the computer screen at work). To set up a secure Remote Desktop connection you must enable Remote Desktop on the target Windows computer that is to be accessed and configure the RPD client software on the client PC.

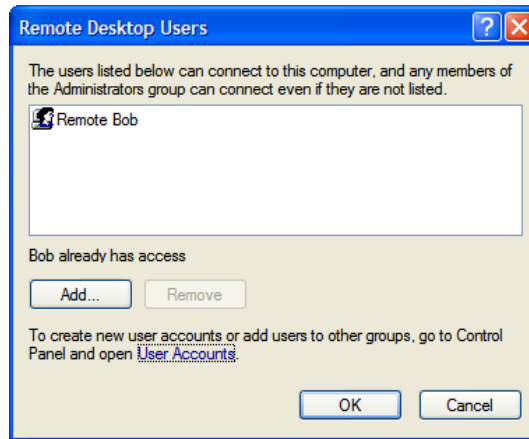
6.8.1 Enable Remote Desktop on the target Windows computer to be accessed

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

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



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



- 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). So more than one user can have active sessions 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 your computer at work, you can unlock it by typing CTRL+ALT+DEL.

6.8.2 Configure the Remote Desktop Connection client

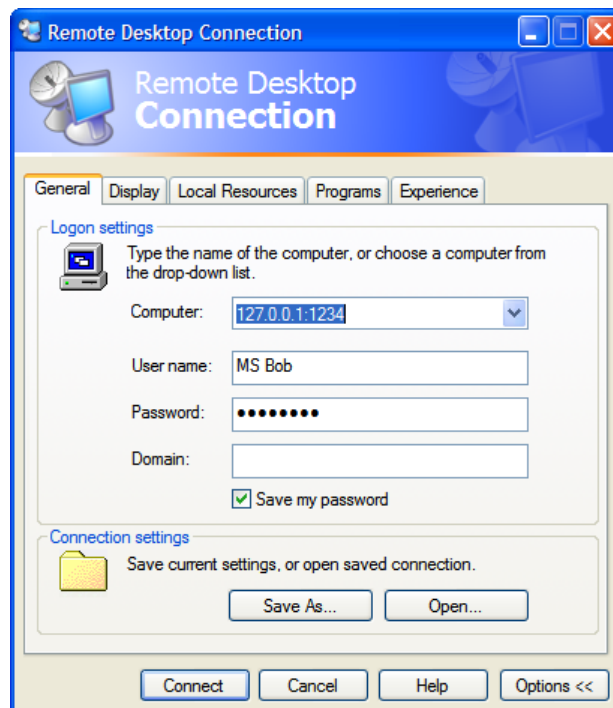
Now you have the Client PC securely connected to the *console server* (either locally, or remotely - thru the enterprise VPN, or a secure SSH internet tunnel or a dial-in SSH tunnel) you can establish the Remote Desktop connection from the Client. To do this you simply enable the **Remote Desktop Connection**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

A. On a Windows client PC

- 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:
 - 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* serial port that is attached to the Windows computer to be controlled e.g. 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
 - 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)



- Click **Connect**

Note The Remote Desktop Connection software is pre-installed with Windows XP and later however for earlier Windows PCs 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 and Windows 2000. When run, this software allows these older Windows platforms to remotely connect to a computer running current Windows.

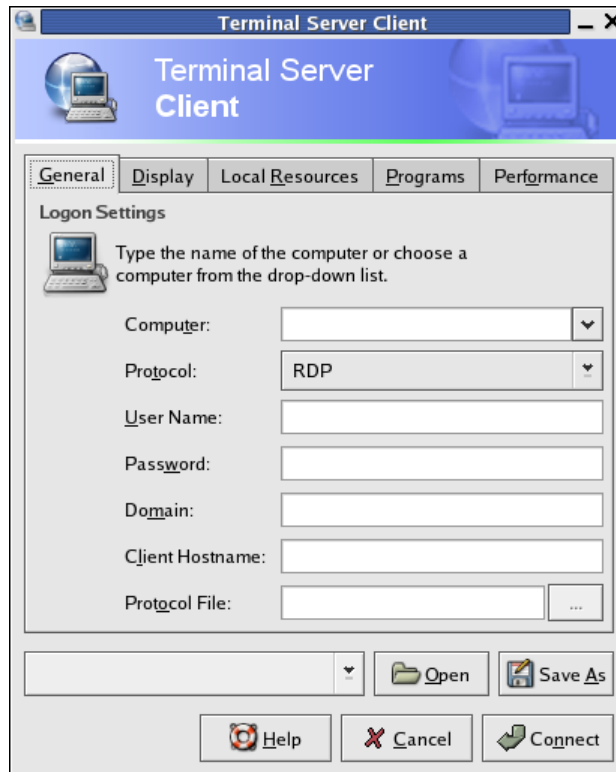
B. On a Linux or UNIX client PC:

- 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: <i>widthxheight</i> 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)



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 you must install and configure the VNC Server software on the computer to be accessed, then install and configure the VNC Viewer software on the Viewer PC.

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 PC, 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 and they are generally can be launched from the (Gnome/KDE etc) front end e.g. with Red Hat Enterprise Linux 4 there's VNC Server software and a choice of Viewer client software, and 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



- To set up a persistent VNC server on Red Hat Enterprise Linux 4:
 - Set a password using **vncpasswd**
 - Edit **/etc/sysconfig/vncservers**
 - Enable the service with **chkconfig vncserver on**
 - Start the service with **service vncserver start**
 - 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 the 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

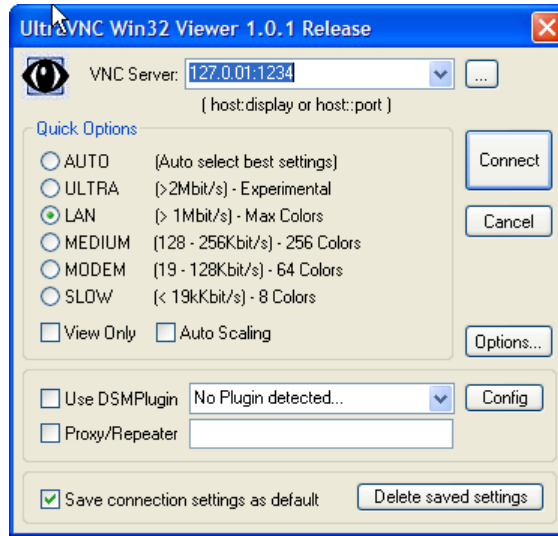
Note To make VNC faster, when you set up the Viewer:

- 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

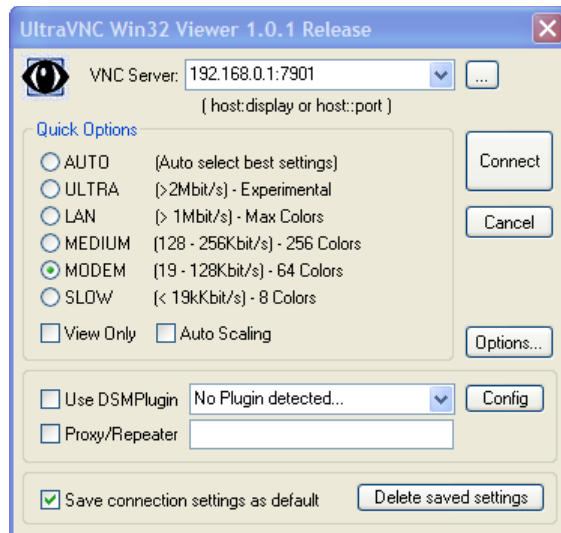
(Refer to <http://doc.uvnc.com> for detailed configuration instructions)

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

A. When the Viewer PC is connected to the *console server* through a 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*



- B. When the Viewer PC is connected directly to the *console server* (i.e. locally or remotely through a VPN or dial in connection); and the VNC Host computer is serially connected to the *console server*; enter the IP address of the *console server* unit with the TCP port that the SDT tunnel will use. The TCP port will be 7900 plus the physical serial port number (i.e. 7901 to 7948, so all traffic directed to port 79xx on the *console server* is tunneled thru to port 5900 on the PPP connection on serial Port xx) e.g. for a Windows Viewer PC using UltraVNC connecting to a VNC Server which is attached to Port 1 on a *console server* located 192.168.0.1



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



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/remotedintro.msp>
 - *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.msp>
 - *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 Using SDT to IP connect to hosts that are serially attached to the gateway

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

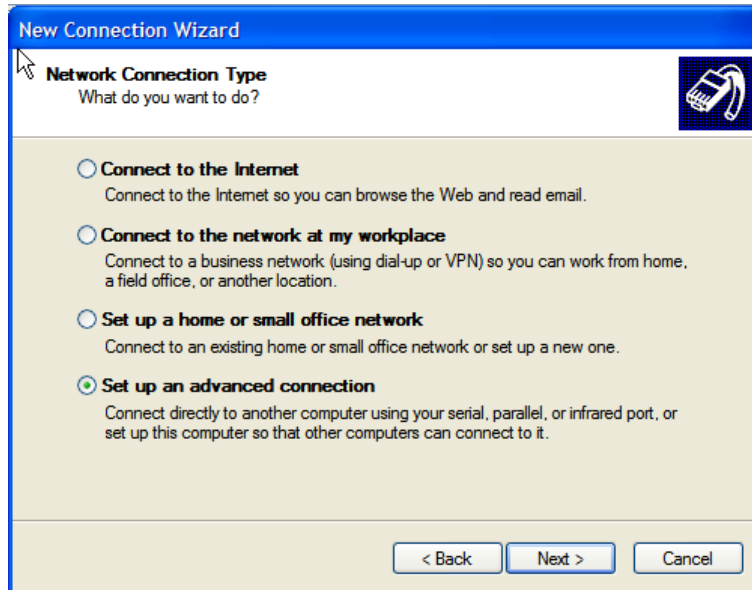
- establish a PPP connection (Section 6.7.1) between the host and the gateway, then
- set up Secure Tunneling - Ports on the *console server* (Section 6.7.2), then
- configure *SDT Connector* to use the appropriate network protocol to access IP consoles on the host devices that are attached to the *Console server* serial ports (Section 6.7.3)

6.10.1 Establish a PPP connection between the host COM port and *console server*

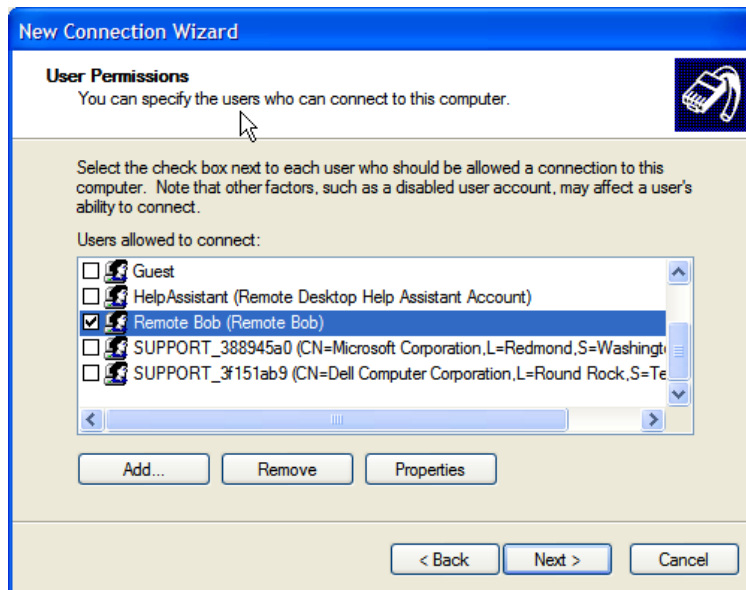
(This step is only necessary for serially connected computers)

Firstly, physically connect the COM port on the host computer that is to be accessed, to the serial port on the *console server* then:

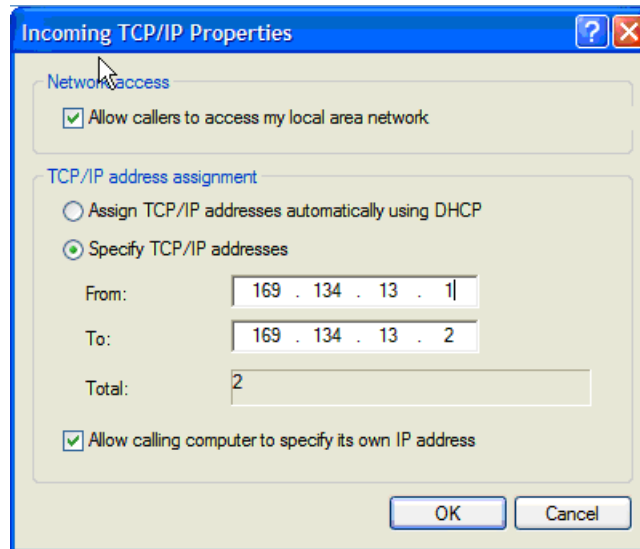
- A. For non Windows (Linux, UNIX, Solaris etc) computers establish a PPP connection over the serial port. The online tutorial <http://www.yolinux.com/TUTORIALS/LinuxTutorialPPP.html> presents a selection of methods for establishing a PPP connection for Linux
- B. For Windows XP and 2003 computers follow 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**



- 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**



- 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**



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

Note You can choose any TCP/IP addresses so 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*

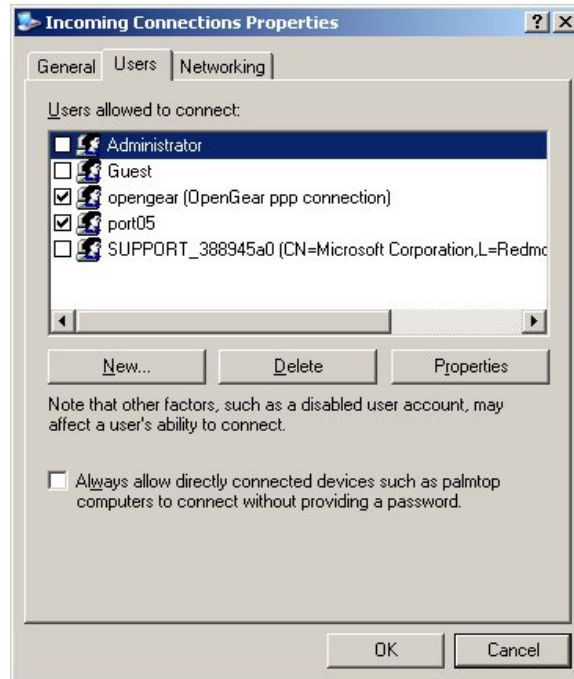
Also you could use the *console server* default username and password when you set up the new Remote Desktop *User* and gave this *User* permission to use the advance connection to access the Windows computer:

- 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 a 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 similar for later versions however the set up screens present slightly differently:



You need to put a check in the box for *Always allow directly connected devices such as palmtop.....*

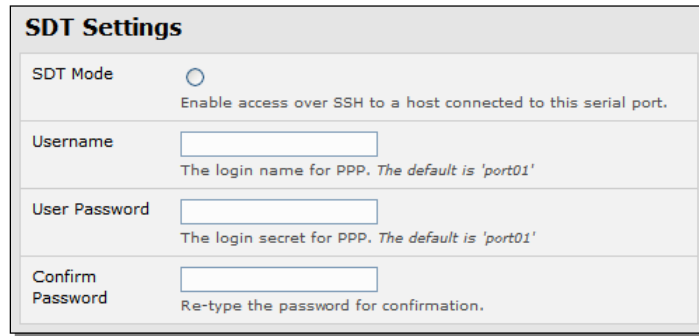
Also the option for to **Set up an advanced connection** is not available in Windows 2003 if RRAS is configured. If RRAS has been configured it is a simply task to enable the null modem connection for the dial-in configuration.

- C. For earlier version Windows computers again follow the steps in Section B. above, however to get to the **Make New Connection** button:
- For Windows 2000, click **Start** and select **Settings** then at the **Dial-Up Networking Folder** click **Network and Dial-up Connections** and click **Make New Connection**. Note you may need to first set up 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* 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**.



SDT Settings	
SDT Mode	<input type="radio"/> Enable access over SSH to a host connected to this serial port.
Username	<input type="text"/> The login name for PPP. The default is 'port01'
User Password	<input type="text"/> The login secret for PPP. The default is 'port01'
Confirm Password	<input type="text"/> Re-type the password for confirmation.

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.

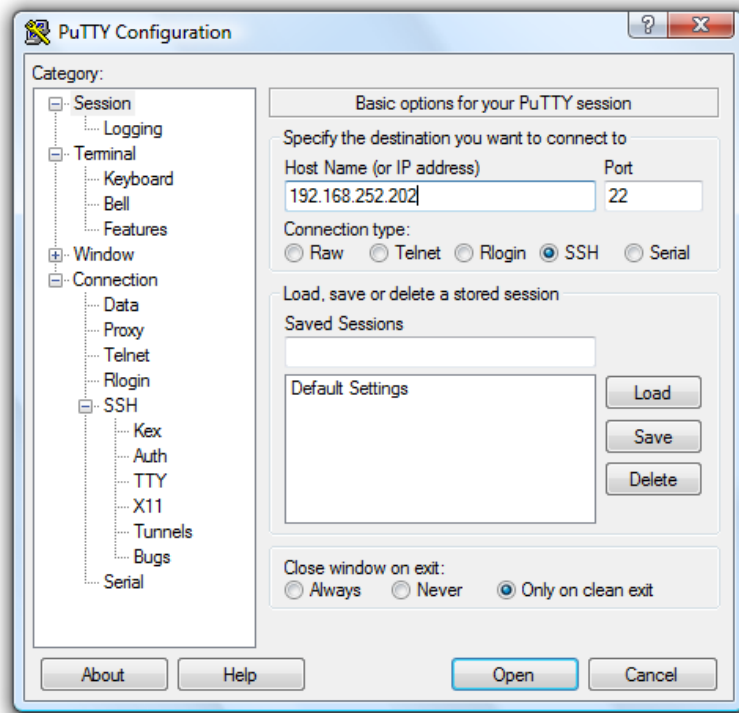
Next you need to add a New SDT Host. In the Host address you need to put portxx where xx = the port you are connecting to. Example for port 3 you would have a Host Address of: port03 and then select the RDP Service check box.

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

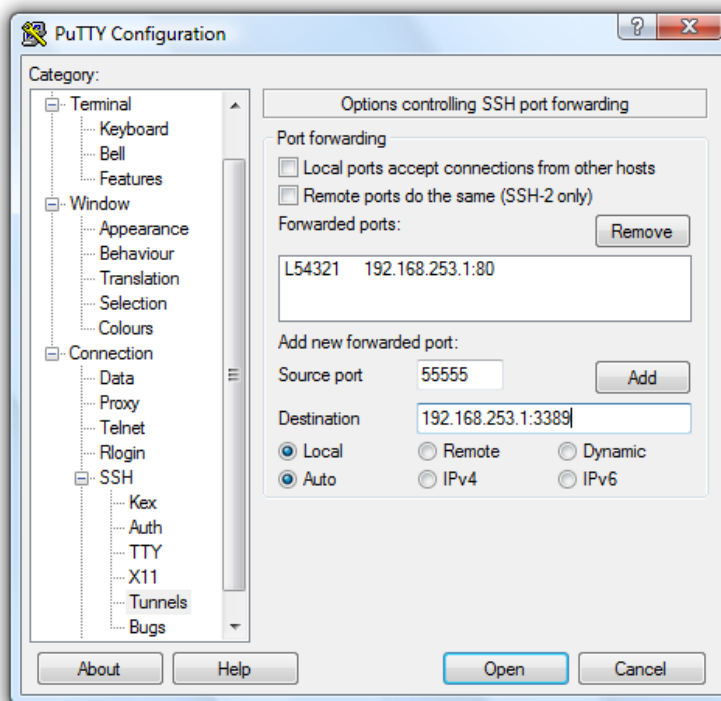
As covered in the previous sections of this chapter we recommend you use the *SDT Connector* client software that is supplied with the *console server*. However there's also a wide selection of commercial and free SSH client programs that can also provide the secure SSH connections to the *console servers*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



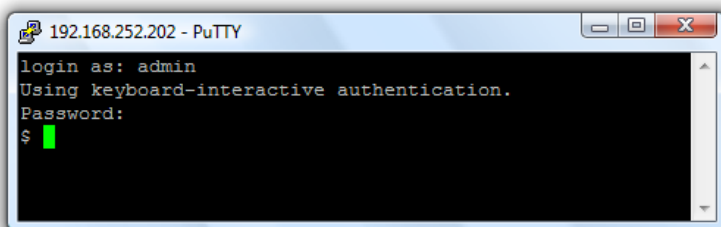
- In the **Session** menu enter the IP address of the *console server* in the **Host Name or IP address** field
 - For dial-in connections, this IP address will be the **Local** Address that you assigned to the *console server* when you set it up as the Dial-In PPP Server
 - For Internet (or local/VPN connections) 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**
 - 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 *192.168.253.1:3389* (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).



- 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



- 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 (which 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.

APPENDIX A: Linux Commands & Source Code

The *console server* platform is a dedicated Linux computer, optimized to provide monitoring and secure access to serial and network consoles of critical server systems and their supporting power and networking infrastructure.

Opengear *console servers* are built on the 2.6 uCLinux kernel as developed by the uCLinux project (except for SD4001/4002 which have less flash and use 2.4 uCLinux kernel). This is GPL code and source can be found at <http://cvs.uclinux.org>.

Some uCLinux commands have config files that can be altered (e.g. *portmanager*, *inetd*, *init*, *ssh/sshd/scp/sshkeygen*, *ucd-snmpd*, *samba*, *fnord*, *sslwrap*).

Other commands you can run and do neat stuff with (e.g. *loopback*, *bash (shell)*, *ftp*, *hwclock*, *iproute*, *iptables*, *netcat*, *ifconfig*, *mii-tool*, *netstat*, *route*, *ping*, *portmap*, *pppd*, *routed*, *setserial*, *smtpclient*, *stty*, *stunel*, *tcpdump*, *tftp*, *tip*, *traceroute*)

Below are most of the standard uCLinux and Busybox commands (and some custom Opengear commands) that are in the default build tree. The *Administrator* can use these to configure the *console server*, and monitor and manage attached serial console and host devices:

addgroup *	Add a group or add an user to a group
adduser *	Add an user
agetty	alternative Linux getty
arp	Manipulate the system ARP cache
arping	Send ARP requests/replies
bash	GNU Bourne-Again Shell
busybox	Swiss army knife of embedded Linux commands
cat *	Concatenate FILE(s) and print them to stdout
chat	Useful for interacting with a modem connected to stdin/stdout
chgrp *	Change file access permissions
chmod *	Change file access permissions
chown *	Change file owner and group
config	Opengear tool to manipulate and query the system configuration from the command line
cp *	Copy files and directories
date *	Print or set the system date and time
dd *	Convert and copy a file
deluser *	Delete USER from the system
df *	Report file system disk space usage
dhcpcd	Dynamic Host Configuration Protocol server
discard	Network utility that listens on the discard port
dmesg *	Print or control the kernel ring buffer
echo *	Print the specified ARGs to stdout
erase	Tool for erasing MTD partitions
eraseall	Tool for erasing entire MTD partitions
false *	Do nothing, unsuccessful
find	Search for files
flashw	Write data to individual flash devices
flatfsd	Daemon to save RAM file systems back to FLASH
ftp	Internet file transfer program
gen-keys	SSH key generation program

getopt *	Parses command options
gettyd	Getty daemon
grep *	Print lines matching a pattern
gunzip *	Compress or expand files
gzip *	Compress or expand files
hd	ASCII, decimal, hexadecimal, octal dump
hostname *	Get or set hostname or DNS domain name
httpd	Listen for incoming HTTP requests
hwclock	Query and set hardware clock (RTC)
inetd	Network super-server daemon
inetd-echo	Network echo utility
init	Process control initialization
ip	Show or manipulate routing, devices, policy routing and tunnels
ipmitool	Linux IPMI manager
iptables	Administration tool for IPv4 packet filtering and NAT
ip6tables	Administration tool for IPv6 packet filtering
iptables-restore	Restore IP Tables
iptables-save	Save IP Tables
kill *	Send a signal to a process to end gracefully
ln *	Make links between files
login	Begin session on the system
loopback	Opengear loopback diagnostic command
loopback1	Opengear loopback diagnostic command
loopback2	Opengear loopback diagnostic command
loopback8	Opengear loopback diagnostic command
loopback16	Opengear loopback diagnostic command
loopback48	Opengear loopback diagnostic command
ls *	List directory contents
mail	Send and receive mail
mkdir *	Make directories
mkfs.jffs2	Create an MS-DOS file system under Linux
mknod *	Make block or character special files
more *	File perusal filter for crt viewing
mount *	Mount a file system
msmtp	SMTP mail client
mv *	Move (rename) files
nc	TCP/IP Swiss army knife
netflash	Upgrade firmware on uLinux platforms using the blkmem interface
netstat	Print network connections, routing tables, interface statistics etc
ntpd	Network Time Protocol (NTP) daemon
pgrep	Display process(es) selected by regex pattern
pidof	Find the process ID of a running program
ping	Send ICMP ECHO_REQUEST packets to network hosts
ping6	IPv6 ping
pkill	Sends a signal to process(es) selected by regex pattern
pmchat	Opengear command similar to the standard chat command (via portmanager)
pmdeny	

pminetd	
pmloggerd	
pmshell	Opengear command similar to the standard <i>tip</i> or <i>cu</i> but all serial port access is directed via the portmanager.
pmusers	Opengear command to query portmanager for active user sessions
portmanager	Opengear command that handles all serial port access
portmap	DARPA port to RPC program number mapper
pppd	Point-to-Point protocol daemon
ps *	Report a snapshot of the current processes
pwd *	Print name of current/working directory
reboot *	<i>Soft</i> reboot
rm *	Remove files or directories
rmdir *	Remove empty directories
routed	Show or manipulate the IP routing table
routed	Show or manipulate the IP routing table
routef	IP Route tool to flush IPv4 routes
routel	IP Route tool to list routes
rtacct	Applet printing /proc/net/rt_acct
rtmon	RTnetlink listener
scp	Secure copy (remote file copy program)
sed *	Text stream editor
setmac	Sets the MAC address
setserial	Sets and reports serial port configuration
sh	Shell
showmac	Shows MAC address
sleep *	Delay for a specified amount of time
smbmnt	Helper utility for mounting SMB file systems
smbmount	Mount an SMBFS file system
smbumount	SMBFS umount for normal users
snmpd	SNMP daemon
snmptrap	Sends an SNMP notification to a manager
sredird	RFC 2217 compliant serial port redirector
ssh	OpenSSH SSH client (remote login program)
ssh-keygen	Authentication key generation, management, and conversion
sshd	OpenSSH SSH daemon
sslwrap	Program that allows plain services to be accessed via SSL
stty	Change and print terminal line settings
stunnel	Universal SSL tunnel
sync *	Flush file system buffers
sysctl	Configure kernel parameters at runtime
syslogd	System logging utility
tar *	The tar archiving utility
tc	Show traffic control settings
tcpdump	Dump traffic on a network
telnetd	Telnet protocol server
tftp	Client to transfer a file from/to tftp server
tftpd	Trivial file Transfer Protocol (tftp) server
tip	Simple terminal emulator/cu program for connecting to modems and serial devices
top	Provide a view of process activity in real time

touch *	Change file timestamps
traceroute	Print the route packets take to network host
traceroute6	Traceroute for IPv6
true *	Returns an exit code of TRUE (0)
umount *	Unmounts file systems
uname *	Print system information
usleep *	Delay for a specified amount of time
vconfig *	Create and remove virtual Ethernet devices
vi *	Busybox clone of the VI text editor
w	Show who is logged on and what they are doing
zcat *	Identical to gunzip -c

Commands above which are appended with '*' come from Busybox (the Swiss Army Knife of embedded Linux) <http://www.busybox.net/downloads/BusyBox.html>.

Others are generic Linux commands and most commands the **-h** or **--help** argument to provide a terse runtime description of their behavior. More details on the generic Linux commands can found online at <http://en.tldp.org/HOWTO/HOWTO-INDEX/howtos.html> and <http://www.faqs.org/docs/Linux-HOWTO/Remote-Serial-Console-HOWTO.html>

An updated list of the commands in the latest *console server* build can be found at <http://www.opengear.com/faq233.html>. However it may be worth using **ls** command to view all the commands actually available in the */bin* directory in your *console server*.

There were a number of Opengear tools listed above that make it simple to configure the *console server* and ensure the changes are stored in the *console server's* flash memory etc. These commands are covered in the previous chapters and include:

- **config** which allows manipulation and querying of the system configuration from the command line. With *config* a new configuration can be activated by running the relevant configurator, which performs the action necessary to make the configuration changes live
- **portmanager** which provides a buffered interface to each serial port. It is supported by the *pmchat* and *pmshell* commands which ensure all serial port access is directed via the *portmanager*
- **pmpower** is a configurable tool for manipulating remote power devices that are serially or network connected to the *console server*
- **SDT Connector** is a java client applet that provides point-and-click SSH tunneled connections to the *console server* and Managed Devices

There are also a number of other CLI commands related to other open source tools embedded in the *console server* including:

- **PowerMan** provides power management for many preconfigured remote power controller (RPC) devices. For CLI details refer <http://linux.die.net/man/1/powerman>
- **Network UPS Tools (NUT)** provides reliable monitoring of UPS and PDU hardware and ensure safe shutdowns of the systems which are connected - with a goal to monitor every kind of UPS and PDU. For CLI details refer <http://www.networkupstools.org>
- **Nagios** is a popular enterprise-class management tool that provides central monitoring of the hosts and services in distributed networks. For CLI details refer <http://www.nagios.org>

Many components of the *console server* software are licensed under the GNU General Public License (version 2), which Opengear supports. You may obtain a copy of the GNU General Public License at <http://www.fsf.org/copyleft/gpl.html>. Opengear will provide source code for any of the components of the software licensed under the GNU General Public License upon request.

Note The software included in each Opengear console server contains copyrighted software that is licensed under the GPL (refer Appendix F for a copy of the GPL license). You may obtain the latest snapshot source code package on a CD by sending a money order or check for \$5 to:

Opengear Support
630 West 9560 South, Suite A
Sandy, UT 84070, USA

Alternately the complete source code corresponding to each released version is available from us for a period of three years after its last shipment. If you would like the source code for an earlier release than the latest current release please write "source for firmware Version x.xx " in the memo line of your payment.

This offer is valid to anyone in receipt of this information.

The *console server* also embodies the *okvm* console management software. This is GPL code and the full source is available from <http://okvm.sourceforge.net>.

The *console server* BIOS (boot loader code) is a port of *uboot* which is also a GPL package with source openly available.

The *console server* CGIs (the html code, xml code and web config tools for the Management Console) are proprietary to Opengear, however the code will be provided to customers, under NDA.

Also inbuilt in the *console server* is a Port Manager application and Configuration tools as described in *Chapters 14* and *15*. These both are proprietary to Opengear, but open to customers (as above).

The *console server* also supports GNU *bash* shell script enabling the *Administrator* to run custom scripts. GNU *bash*, version 2.05.0(1)-release (arm-OpenGear-linux-gnu) offers the following shell commands:

alias [-p] [name[=value] ...]	local name[=value] ...
bg [job_spec]	logout
bind [-lpvsPVS] [-m keymap] [-f fi	popd [+N -N] [-n]
break [n]	printf format [arguments]
builtin [shell-builtin [arg ...]]	pushd [dir +N -N] [-n]
case WORD in [PATTERN [pwd [-PL]
PATTERN]	read [-ers] [-t timeout] [-p prompt]
cd [-PL] [dir]	readonly [-anf] [name ...] or read return
command [-pVv]	[n]
command [arg ...]	select NAME [in WORDS ... ;] do
compgen [-abcdefjkvu] [-o option]	COMMANDS
complete [-abcdefjkvu] [-pr] [-o o]	set [--abefhkmnptuvxBCHP] [-o opti]
continue [n]	shift [n]
declare [-afFrxi] [-p] name[=value]	shopt [-pqsu] [-o long-option] opt
dirs [-clpv] [+N] [-N]	source filename
disown [-h] [-ar] [jobspec ...]	suspend [-f]
echo [-neE] [arg ...]	test [expr]
enable [-pnds] [-a] [-f filename]	time [-p] PIPELINE
eval [arg ...]	times
exec [-cl] [-a name] file [redirec]	trap [arg] [signal_spec ...]
exit [n]	true
export [-nf] [name ...] or export	type [-apt] name [name ...]
false	typeset [-afFrxi] [-p] name[=value]
fc [-e ename] [-nlr] [first] [last]	ulimit [-SHacdflmnpstuv] [limit]
fg [job_spec]	umask [-p] [-S] [mode]

<pre>for NAME [in WORDS ... ;] do COMMA function NAME { COMMANDS ; } or NA getopts optstring name [arg] hash [-r] [-p pathname] [name ...] help [-s] [pattern ...] history [-c] [-d offset] [n] or hi if COMMANDS; then COMMANDS; [elif jobs [-lnprs] [jobspec ...] or <i>job kill</i> [-s sigspec -n signum -si let arg [arg ...]</pre>	<pre>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 ; }</pre>
---	--

APPENDIX B: TERMINOLOGY

TERM	MEANING
3G	Third-generation cellular technology. The standards that determine 3G call for greater bandwidth and higher speeds for cellular networks
AES	The Advanced Encryption Standard (AES) is a new block cipher standard to replace DES, developed by NIST, the US National Institute of Standards and Technology. AES ciphers use a 128-bit block and 128-, 192-, or 256-bit keys. The larger block size helps resist birthday attacks while the large key size prevents brute force attacks.
APN	Access Point Name (APN) is used by carriers to identify an IP packet data network that a mobile data user wants to communicate with and the type of wireless service
Authentication	Authentication is the technique by which a process verifies that its communication partner is who it is supposed to be and not an imposter. Authentication confirms that data is sent to the intended recipient and assures the recipient that the data originated from the expected sender and has not been altered on route
BIOS	Basic Input/Output System is the built-in software in a computer that are executed on startup (boot) and that determine what the computer can do without accessing programs from a disk. On PCs, the BIOS contains all the code required to control the keyboard, display screen, disk drives, serial communications, and a number of miscellaneous functions
Bonding	Ethernet Bonding or Failover is the ability to detect communication failure transparently, and switch from one LAN connection to another.
BOOTP	Bootstrap Protocol. A protocol that allows a network user to automatically receive an IP address and have an operating system boot without user interaction. BOOTP is the basis for the more advanced DHCP
Certificates	A digitally signed statement that contains information about an entity and the entity's public key, thus binding these two pieces of information together. A certificate is issued by a trusted organization (or entity) called a Certification Authority (CA) after the CA has verified that the entity is who it says it is.
Certificate Authority	A Certificate Authority is a trusted third party, which certifies public key's to truly belong to their claimed owners. It is a key part of any Public Key Infrastructure, since it allows users to trust that a given public key is the one they wish to use, either to send a private message to its owner or to verify the signature on a message sent by that owner.
Certificate Revocation List	A list of certificates that have been revoked by the CA before they expired. This may be necessary if the private key certificate has been compromised or if the holder of the certificate is to be denied the ability to establish a connection to the <i>console server</i> .
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.
CMS	The term <i>CMS</i> refers to the Centralized Management Software running in all the <i>Lighthouse</i> appliances
Console server	The term <i>console server</i> refers generically to the Opendgear datacenter and remote management appliances, including the ACM5000, ACM5500, IM4200, CM41000 and SD4000 product lines.
DES	The Data Encryption Standard is a block cipher with 64-bit blocks and a 56-bit key.
DHCP	Dynamic Host Configuration Protocol. A communications protocol that assigns IP addresses to computers when they are connected to the network.

DNS	Domain Name System that allocates Internet domain names and translates them into IP addresses. A domain name is a meaningful and easy to remember name for an IP address.
DUN	Dial Up Networking
Encryption	The technique for converting a readable message (plaintext) into apparently random material (ciphertext) which cannot be read if intercepted. The proper decryption key is required to read the message.
Ethernet	A physical layer protocol based upon IEEE standards
Firewall	A network gateway device that protects a private network from users on other networks. A firewall is usually installed to allow users on an intranet access to the public Internet without allowing public Internet users access to the intranet.
Gateway	A machine that provides a route (or pathway) to the outside world.
Hub	A network device that allows more than one computer to be connected as a LAN, usually using UTP cabling.
Internet	A worldwide system of computer networks - a public, cooperative, and self-sustaining network of networks accessible to hundreds of millions of people worldwide. The Internet is technically distinguished because it uses the TCP/IP set of protocols.
Intranet	A private TCP/IP network within an enterprise.
IPMI	Intelligent Platform Management Interface (IPMI) is a set of common interfaces to a computer system which system administrators can use to monitor system health and manage the system. The IPMI standard defines the protocols for interfacing with a service processor embedded into a server platform.
Key lifetimes	The length of time before keys are renegotiated
LAN	Local Area Network
LDAP	The Lightweight Directory Access Protocol (LDAP) is based on the X.500 standard, but significantly simpler and more readily adapted to meet custom needs. The core LDAP specifications are all defined in RFCs. LDAP is a protocol used to access information stored in an LDAP server.
LED	Light-Emitting Diode
<i>Lighthouse</i> appliance	This term refers generically to the <i>Lighthouse</i> VM software appliance, the <i>Lighthouse</i> Standard hardware appliance and <i>Lighthouse</i> Enterprise hardware appliance
MAC address	Every piece of Ethernet hardware has a unique number assigned to it called its MAC address. Ethernet is used locally to connect the <i>console server</i> to the Internet, and it may share the local network with many other appliances. The MAC address is used by the local Internet router in order to direct <i>console server</i> traffic to it rather than somebody else in the local area. It is a 48-bit number usually written as a series of 6 hexadecimal octets, e.g. 00:d0:cf:00:5b:da. A <i>console server</i> has a MAC address listed on a label underneath the device.
Managed Console Server	<i>Managed Console Server</i> refers generically to any <i>console server</i> that is being centrally managed by a <i>Lighthouse</i> appliance.
MSCHAP	Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is authentication for PPP connections between a computer using a Microsoft Windows operating system and a network access server. It is more secure than PAP or CHAP, and is the only option that also supports data encryption.

NAT	Network Address Translation. The translation of an IP address used on one network to an IP address on another network. Masquerading is one particular form of NAT.
Net mask	The way that computers know which part of a TCP/IP address refers to the network, and which part refers to the host range.
NFS	Network File System is a protocol that allows file sharing across a network. Users can view, store, and update files on a remote computer.
NTP	Network Time Protocol (NTP) used to synchronize clock times in a network of computers
OUT OF BAND	Out-of-Band (OOB) management is any management done over channels and interfaces that are separate from those used for user/customer data. Examples would include a serial console interface or a network interface connected to a dedicated management network that is not used to carry customer traffic, or to a BMC/service processor. Any management done over the same channels and interfaces used for user/customer data is In Band.
PAP	Password Authentication Protocol (PAP) is the usual method of user authentication used on the internet: sending a username and password to a server where they are compared with a table of authorized users. Whilst most common, PAP is the least secure of the authentication options.
PPP	Point-to-Point Protocol. A networking protocol for establishing simple links between two peers.
RADIUS	The Remote Authentication Dial-In User Service (RADIUS) protocol was developed by Livingston Enterprises as an access server authentication and accounting protocol. The RADIUS server can support a variety of methods to authenticate a user. When it is provided with the username and original password given by the user, it can support PPP, PAP or CHAP, UNIX login, and other authentication mechanisms.
Router	A network device that moves packets of data. A router differs from hubs and switches because it is "intelligent" and can route packets to their final destination.
SIM	Subscriber Identity Module (SIM) card stores unique serial numbers and security authentication used to identify a subscriber on mobile telephony devices
SMASH	Systems Management Architecture for Server Hardware is a standards-based protocols aimed at increasing productivity of the management of a data center. The SMASH Command Line Protocol (SMASH CLP) specification provides an intuitive interface to heterogeneous servers independent of machine state, operating system or OS state, system topology or access method. It is a standard method for local and remote management of server hardware using out-of-band communication
SMTP	Simple Mail Transfer Protocol. <i>console server</i> includes, SMTPclient, a minimal SMTP client that takes an email message body and passes it on to a SMTP server (default is the MTA on the local host).
SOL	Serial Over LAN (SOL) enables servers to transparently redirect the serial character stream from the baseboard universal asynchronous receiver/transmitter (UART) to and from the remote-client system over a LAN. With SOL support and BIOS redirection (to serial) remote managers can view the BIOS/POST output during power on, and reconfigured.
SSH	Secure Shell is secure transport protocol based on public-key cryptography.
SSL	Secure Sockets Layer is a protocol that provides authentication and encryption services between a web server and a web browser.
TACACS+	The Terminal Access Controller Access Control System (TACACS+) security protocol is a more recent protocol developed by Cisco. It provides detailed accounting information and flexible administrative control over the

	authentication and authorization processes. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide authentication, authorization, and accounting services independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon. There is a draft RFC detailing this protocol.
TCP/IP	Transmission Control Protocol/Internet Protocol. The basic protocol for Internet communication.
TCP/IP address	Fundamental Internet addressing method that uses the form nnn.nnn.nnn.nnn.
Telnet	Telnet is a terminal protocol that provides an easy-to-use method of creating terminal connections to a network.
UDP	User Datagram Protocol
UTC	Coordinated Universal Time.
UTP	Unshielded Twisted Pair cabling. A type of Ethernet cable that can operate up to 100Mb/s. Also known as Category 5 or CAT 5.
VNC	Virtual Network Computing (VNC) is a desktop protocol to remotely control another computer. It transmits the keyboard presses and mouse clicks from one computer to another relaying the screen updates back in the other direction, over a network.
VPN	Virtual Private Network (VPN) a network that uses a public telecommunication infrastructure and Internet, to provide remote offices or individual users with secure access to their organization's network
WAN	Wide Area Network
WINS	Windows Internet Naming Service (WINS) that manages the association of workstation names and locations with IP addresses

APPENDIX C: End User License Agreement (EULA)

Lighthouse hardware appliance EULA



Read before using the Lighthouse Standard or Enterprise hardware appliance

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE USING THE LIGHTHOUSE HARDWARE APPLIANCE, 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 APPLIANCE. IF YOU USE ANY PART OF THE APPLIANCE, SUCH USE WILL INDICATE THAT YOU ACCEPT THESE TERMS.

You have acquired a product that includes Opendgear (“Opendgear”) proprietary software and/or proprietary software licensed to Opendgear. This Opendgear End User License Agreement (“EULA”) is a legal agreement between you (either an individual or a single entity) and Opendgear for the installed software product of Opendgear 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, Opendgear 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, Opendgear grants you a nonexclusive right and license to install and use the Software on a single physical 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;
- (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;
- (4) you may not use the Software to concurrently manage more than the number of appliances designated when you purchased with your Lighthouse hardware appliance (i.e. 10, 100, 1000 or 5000 appliances)

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. Opendgear 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. Opendgear 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 Opendgear supports. Opendgear will provide source code for any of the components of the Software licensed under the GNU General Public License upon request.
- (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 on the [Opendgear web site](#),

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 license you purchased with your Lighthouse hardware appliance has an associated active term (i.e. one year or three years). Opengear reserves the right to terminate product support and product updates if the active license term has expired. However this term expiration does not affect the EULA validity. 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 Opengear 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 Opengear for any reason, please contact the Opengear 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, AND OPENGEAR HAS RELIED UPON YOU TO CONDUCT SUFFICIENT TESTING TO DETERMINE THAT THE SOFTWARE IS SUITABLE FOR SUCH USE.

LIMITED WARRANTY

Opengear warrants the media containing the Software for a period of ninety (90) days from the date of original purchase from Opengear or its authorized retailer. Proof of date of purchase will be required. Any updates to the Software provided by Opengear (which may be provided by Opengear at its sole discretion) shall be governed by the terms of this EULA. In the event the product fails to perform as warranted, Opengear's sole obligation shall be, at Opengear'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. Opengear 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.

OPENGEAR 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, OPENGEAR.

NO LIABILITY FOR CERTAIN DAMAGES. EXCEPT AS PROHIBITED BY LAW, OPENGEAR 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 OPENGEAR BE LIABLE FOR ANY AMOUNT IN EXCESS OF THE LICENSE FEE PAID TO OPENGEAR 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.

Lighthouse VM software appliance EULA



Read before using the Lighthouse VM software appliance

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE USING THE LIGHTHOUSE VM SOFTWARE APPLIANCE, 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 Opengear (“Opengear”) proprietary software and/or proprietary software licensed to Opengear. This Opengear End User License Agreement (“EULA”) is a legal agreement between you (either an individual or a single entity) and Opengear for the installed software product of Opengear 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, Opengear 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, Opengear grants you a nonexclusive right and license to install and use the Software on a single physical or virtual CPU, and to install and use the Software on a second physical or virtual CPU to serve as an idle stand-by, 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;
- (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;
- (4) if you have not obtained and installed a Software License Key from Opengear (or its authorized reseller) you are permitted to use the Software solely for evaluation or demonstration purposes however your right to use the Software shall terminate thirty (30) days after your installation of the Software, at which time you must return or destroy the Software; and
- (5) if you have installed a Software License Key you may not use the Software to concurrently manage more than the number of appliances specified in that Software License Key

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. Opengear 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. Opengear 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 Opengear supports. Opengear will provide source code for any of the components of the Software licensed under the GNU General Public License upon request.
- (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 on the [Opengear web site](#),

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. Each unique Lighthouse VM Software license has an associated active term. Opengear reserves the right to terminate product support and product updates if the active license term has expired. However this term expiration does not affect the EULA validity. 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 Opengear 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 Opengear for any reason, please contact the Opengear 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, AND OPENGEAR HAS RELIED UPON YOU TO CONDUCT SUFFICIENT TESTING TO DETERMINE THAT THE SOFTWARE IS SUITABLE FOR SUCH USE.

LIMITED WARRANTY

Opengear warrants the media containing the Software for a period of ninety (90) days from the date of original purchase from Opengear or its authorized retailer. Proof of date of purchase will be required. Any updates to the Software provided by Opengear (which may be provided by Opengear at its sole discretion) shall be governed by the terms of this EULA. In the event the product fails to perform as warranted, Opengear's sole obligation shall be, at Opengear'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. Opengear 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.

OPENGEAR 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, OPENGEAR.

NO LIABILITY FOR CERTAIN DAMAGES. EXCEPT AS PROHIBITED BY LAW, OPENGEAR 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 OPENGEAR BE LIABLE FOR ANY AMOUNT IN EXCESS OF THE LICENSE FEE PAID TO OPENGEAR 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.

