Lighthouse User Manual

Lighthouse VM
Lighthouse Standard
Lighthouse Enterprise

Pre-release Rev: 1.0
February 6th 2013
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

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Update details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 2013</td>
<td>1.0</td>
<td>Release manual for Lighthouse VM appliance. Software V4.0</td>
</tr>
</tbody>
</table>

Copyright

©Opengear Inc. 2013. 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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>THIS MANUAL</td>
<td>6</td>
</tr>
<tr>
<td>INITIAL DEPLOYMENT</td>
<td>8</td>
</tr>
<tr>
<td>2.1 Lighthouse VM Deployment</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Lighthouse Standard Deployment</td>
<td>11</td>
</tr>
<tr>
<td>2.3 Lighthouse Enterprise Deployment</td>
<td>11</td>
</tr>
<tr>
<td>CONFIGURATION</td>
<td>12</td>
</tr>
<tr>
<td>3.1 Welcome</td>
<td>12</td>
</tr>
<tr>
<td>3.2.1 Passwords</td>
<td>12</td>
</tr>
<tr>
<td>3.2.1 Enter Call Home Password</td>
<td>13</td>
</tr>
<tr>
<td>3.2.2 Enter License Key</td>
<td>14</td>
</tr>
<tr>
<td>3.3 Configure Local Network Settings</td>
<td>14</td>
</tr>
<tr>
<td>3.3.1 IPv6 configuration</td>
<td>15</td>
</tr>
<tr>
<td>3.3.2 Dynamic DNS (DDNS) configuration</td>
<td>16</td>
</tr>
<tr>
<td>3.3.3 Static routes</td>
<td>17</td>
</tr>
<tr>
<td>3.4 Configure Managed Console Servers</td>
<td>17</td>
</tr>
<tr>
<td>3.4.1 Connecting to sites on separate private or firewalled networks</td>
<td>20</td>
</tr>
<tr>
<td>3.5 Call Home</td>
<td>22</td>
</tr>
<tr>
<td>3.5.1 Setting up console server as a management candidate on CMS</td>
<td>22</td>
</tr>
<tr>
<td>3.5.2 Call Home to a generic central SSH server</td>
<td>23</td>
</tr>
<tr>
<td>3.6 Authorize Automatically Added Users</td>
<td>24</td>
</tr>
<tr>
<td>3.7 Upgrade Firmware</td>
<td>25</td>
</tr>
<tr>
<td>3.8 Configure Date and Time</td>
<td>26</td>
</tr>
<tr>
<td>3.9 Key Exchange</td>
<td>27</td>
</tr>
<tr>
<td>3.10 Authentication Configuration</td>
<td>27</td>
</tr>
<tr>
<td>3.10.1 Local authentication</td>
<td>28</td>
</tr>
<tr>
<td>3.10.2 TACACS authentication</td>
<td>28</td>
</tr>
<tr>
<td>3.10.3 RADIUS authentication</td>
<td>30</td>
</tr>
<tr>
<td>3.10.4 LDAP authentication</td>
<td>30</td>
</tr>
<tr>
<td>3.10.5 Group support with remote authentication</td>
<td>31</td>
</tr>
<tr>
<td>3.10.6 Idle timeout</td>
<td>32</td>
</tr>
<tr>
<td>3.10.7 Authentication testing</td>
<td>33</td>
</tr>
<tr>
<td>3.11 SSL Certificate</td>
<td>33</td>
</tr>
<tr>
<td>3.12 Services and Service Access</td>
<td>35</td>
</tr>
<tr>
<td>3.13 Support Report</td>
<td>36</td>
</tr>
<tr>
<td>3.14 System Reset</td>
<td>37</td>
</tr>
<tr>
<td>3.15 Syslog</td>
<td>37</td>
</tr>
<tr>
<td>3.16 Dialpool</td>
<td>38</td>
</tr>
<tr>
<td>ACCESSING MANAGED CONSOLE SERVERS &amp; DEVICES</td>
<td>42</td>
</tr>
<tr>
<td>4.1 Viewing Managed Console Servers &amp; Devices</td>
<td>42</td>
</tr>
<tr>
<td>4.1.1 Viewing Managed Console Servers</td>
<td>42</td>
</tr>
<tr>
<td>4.1.2 Viewing Managed Devices</td>
<td>43</td>
</tr>
<tr>
<td>4.2 Accessing Managed Console Servers &amp; Devices</td>
<td>44</td>
</tr>
<tr>
<td>4.2.1 Accessing Managed Console Servers</td>
<td>44</td>
</tr>
<tr>
<td>4.2.2 Accessing Managed Devices</td>
<td>45</td>
</tr>
<tr>
<td>4.3 Batch Control Managed Console Servers</td>
<td>46</td>
</tr>
<tr>
<td>4.4 Manage Terminal</td>
<td>48</td>
</tr>
<tr>
<td>MONITORING WITH NAGIOS</td>
<td>50</td>
</tr>
<tr>
<td>5.1 Monitor</td>
<td>50</td>
</tr>
<tr>
<td>5.1.1 Tactical Overview</td>
<td>50</td>
</tr>
<tr>
<td>5.1.3 Services</td>
<td>52</td>
</tr>
<tr>
<td>5.1.4 Problems</td>
<td>52</td>
</tr>
<tr>
<td>5.1.5 Connecting with SDT Connector</td>
<td>53</td>
</tr>
<tr>
<td>5.2 Reports and system</td>
<td>56</td>
</tr>
</tbody>
</table>
5.2.1 Notifications 56

5.3 Extended Nagios 56
5.3.1 Adding custom checks + scripting/config set up 57
5.3.2 Introducing NagVis 57
5.3.3 Notifications 58
5.3.4 Notification Elevation 59

ACCESSING WITH SDT CONNECTOR 64

6.1 Configuring for SSH Tunneling to Hosts 65
6.2 SDT Connector client installation and configuration 65
6.3 SDT Connector to Management Console 74
6.4 SDT Connector - telnet or SSH connect to serially attached devices 75
6.5 Using SDT Connector for out-of-band connection to the gateway 76
6.6 Importing (and exporting) preferences 77
6.7 SDT Connector Public Key Authentication 78
6.8 Setting up SDT for Remote Desktop access 78
6.9 SDT SSH Tunnel for VNC 83
6.10 Using SDT to IP connect to hosts that are serially attached to the gateway 86
6.11 SSH Tunneling using other SSH clients (e.g. PuTTY) 90

APPENDIX A: Linux Commands & Source Code 94
APPENDIX B: TERMINOLOGY 100
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 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, operation, 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  Lighthouse appliance and CMS centralized management software installation
3. Configuration  Initial CMS configuration and connection to the Managed Console Servers
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 – as detailed in Section 11.4.

- The **commit config** icon enables you to commit queue configuration changes – as detailed in Section 11.4.

- Click the **modify** icon to change an 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

![Warning](image)

*Text presented like this highlights important issues and it is essential you read and take head 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.

<table>
<thead>
<tr>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick Start Guide</td>
<td>Leads you through your initial Opengear configuration</td>
</tr>
<tr>
<td>Knowledgebase</td>
<td>opengear.zendesk.com contains a knowledgebase with technical how to articles and tech tips</td>
</tr>
</tbody>
</table>
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 the Lighthouse VM software appliance. Subsequent manual releases will also describe the physical installation of the Lighthouse Standard and Enterprise hardware 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/](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`

2.1.2 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 Opengear 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)
Chapter 2: INITIAL DEPLOYMENT

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 'Commited 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="xxxxxxxxx-xxxx-xxxx-xxxx-
   xxxxxxxxxx:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

   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 VM 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 Deployment

2.3  Lighthouse Enterprise Deployment
Chapter 3: CONFIGURATION

This chapter covers the initial CMS 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 setting Time/Date, upgrading the firmware and connection to the Managed Console Servers

3.1 Welcome

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

- You will arrive at the Welcome to the Opengear Lighthouse VMI 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.

3.2.1 Passwords

- The first step Change the default administration password takes you to Configure: User Authorization, where you can again reset the password for root.

Note: 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, but only the first eight password characters are used to make the password hash.

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

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

![License Key Installation](image)

**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
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 `http://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](http://www.dyns.cx)
  - dyndns.org [www.dyndns.org](http://www.dyndns.org)
  - GNUDip [gnudip.cheapnet.net](http://gnudip.cheapnet.net)
  - ODS [www.ods.org](http://www.ods.org)
  - TZO [www.tzo.com](http://www.tzo.com)
  - 3322.org (Chinese provider) [www.3322.org](http://www.3322.org)

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

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

- 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.

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.
- 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

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").

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 VCMS 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 Portshare gateway). Setting the number of ports also determines how many Ajax Webterms are accessible from the Access Console Server page.
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.

### 3.4.1 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.**

   ![Diagram](image)

   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.**

   ![Diagram](image)

   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:

1. Browse to the console server’s management console and select Call Home on the Serial & Network menu

![Serial & Network: Call Home](image1)

2. 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*

3. Click Add

![Serial & Network: Call Home](image2)

4. Enter the IP address or DNS name (e.g. the dynamic DNS address) of the CMS

5. Enter the Password that you configured on the CMS as the Call Home Password

6. 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

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.

- 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.
3.7 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 VM upgrade files (*.bin) 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.
- 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.
3.8 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.

![Configure Date & Time](image)

- 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**
3.9 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

3.10 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)

### 3.10.1 Local authentication
- Select **Configure: Authentication** and check Local
- Click **Apply**

### 3.10.2 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
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:

3.10.3 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**

- 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.freeradius.org/](http://www.freeradius.org/)

3.10.4 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**
➢ 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/articles/intro_to_ldap.html)
- [http://www.ldapman.org/servers.html](http://www.ldapman.org/servers.html)
- [http://www.linuxplanet.com/linuxplanet/tutorials/5050/1/](http://www.linuxplanet.com/linuxplanet/tutorials/5050/1/)

---

### 3.10.5 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

### 3.10.6 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.

### 3.10.7 Authentication testing

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

![Authentication Testing](image)

### 3.11 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.
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.

3.12 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 console servers.

**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 in plain text format.
Select **Status: Support Report** and you will be presented with a status snapshot

Save the file as a text file and attach it to your support email

### 3.14 System Reset

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

A *soft* reset is affected 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**

### 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

CMS enables you to build modems into a simple virtual modem pool that can be browser accessed by field engineers for out of band connection to remote sites.

The modems in the out-dial pool are serially connected to downstream *Managed Console Servers* - which themselves may be distributed regionally (or internationally).

- To add a modem to the pool select **Configure: Dialpool** and enter the **Host Address** (i.e. the IP address or Domain Name of the downstream *Managed Console Server*) and the **RFC2217 Host Port** address (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 on 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 **Name** and click **Add Modem**
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 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](image)

  - 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.

In the event you wish to access any of these configured console servers through the virtual Dialpool:

- 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
For more detail refer the YouTube training video  http://www.youtube.com/watch?v=HG6jPpGL1sl&feature=youtu.be
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 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.
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**.

### 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 it is a UPS or EMD device etc). For example in the screen below only devices attached to **Managed Console Servers** running V3.6.0b firmware are displayed.
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 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

### 4.3 Batch Control Managed Console Servers

**CMS** can schedule 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.

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

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.

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.
4.4 Manage Terminal

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

- **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 the 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:
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.

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.

![Services Screenshot](image)

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 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.

- 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 to 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.

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 Opengear 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.

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 \texttt{<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 `<contact_groups>` 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](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](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
```

Lighthouse Centralized Management Appliance User Manual
case "$VAR" in
  F ) LOGFILE=$OPTARG ;;
  * ) echo "wrong syntax: use $0 -F <file to check>"
    exit 3 ;;
esac
done

if test "LOGFILE" = ""
then
  echo "wrong syntax: use $0 -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 been 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:

```yaml
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. 7301 to 7348 on a 48 port console server)
- 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:
Chapter 6: ACCESSING WITH SDT CONNECTOR

**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 `sdcon-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/](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 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**)
Chapter 6: ACCESSING WITH SDT CONNECTOR

- 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.
Chapter 6: ACCESSING WITH SDT CONNECTOR

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%`:
Chapter 6: ACCESSING WITH SDT CONNECTOR

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
Chapter 6: ACCESSING WITH SDT CONNECTOR

- 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 is 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:
  \[ \text{cmd /c start "Starting Out of Band Connection" /wait /min rasdial network_connection login password} \]
  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:
  \[ \text{pon network_connection} \]
  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:
  \[ \text{cmd /c start "Stopping Out of Band Connection" /wait /min rasdial network_connection /disconnect} \]
  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:
  \[ \text{poff network_connection} \]

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:
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
- If you do not already have a public/private key pair for your client PC (the one running SDT Connector on) 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://sshwindows.sourceforge.net/download/
- Upload the public part of your SSH key pair (this file is typically named id_rsa.pub or id_dsa.pub) to the SSH gateway, or 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, Vista, Server2003, Server 2008 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, Vista and Server 2003/2008, however for earlier Windows PCs you will need to download the RDP client:

Chapter 6: ACCESSING WITH SDT CONNECTOR

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`

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

- 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)

![Terminal Server Client](image)

**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

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
Chapter 6: ACCESSING WITH SDT CONNECTOR

- Click the **Allow other users...** checkbox to allow remote users to view and control your desktop

![Remote Desktop Preferences](image)

- 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](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](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](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:
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 Vista and Windows Server 2003/2008 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**.
Chapter 6: ACCESSING WITH SDT CONNECTOR

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.

---

**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
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. Alternatively 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, tfp, 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:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addgroup*</td>
<td>Add a group or add an user to a group</td>
</tr>
<tr>
<td>adduser*</td>
<td>Add an user</td>
</tr>
<tr>
<td>agetty</td>
<td>alternative Linux getty</td>
</tr>
<tr>
<td>arp</td>
<td>Manipulate the system ARP cache</td>
</tr>
<tr>
<td>arpping</td>
<td>Send ARP requests/replies</td>
</tr>
<tr>
<td>bash</td>
<td>GNU Bourne-Again Shell</td>
</tr>
<tr>
<td>busybox</td>
<td>Swiss army knife of embedded Linux commands</td>
</tr>
<tr>
<td>cat*</td>
<td>Concatenate FILE(s) and print them to stdout</td>
</tr>
<tr>
<td>chat</td>
<td>Useful for interacting with a modem connected to stdin/stdout</td>
</tr>
<tr>
<td>chgrp*</td>
<td>Change file access permissions</td>
</tr>
<tr>
<td>chmod*</td>
<td>Change file access permissions</td>
</tr>
<tr>
<td>chown*</td>
<td>Change file owner and group</td>
</tr>
<tr>
<td>config</td>
<td>Opengear tool to manipulate and query the system configuration from the command line</td>
</tr>
<tr>
<td>cp*</td>
<td>Copy files and directories</td>
</tr>
<tr>
<td>date*</td>
<td>Print or set the system date and time</td>
</tr>
<tr>
<td>dd*</td>
<td>Convert and copy a file</td>
</tr>
<tr>
<td>deluser*</td>
<td>Delete USER from the system</td>
</tr>
<tr>
<td>df*</td>
<td>Report file system disk space usage</td>
</tr>
<tr>
<td>dhcpd</td>
<td>Dynamic Host Configuration Protocol server</td>
</tr>
<tr>
<td>discard</td>
<td>Network utility that listens on the discard port</td>
</tr>
<tr>
<td>dmesg*</td>
<td>Print or control the kernel ring buffer</td>
</tr>
<tr>
<td>echo*</td>
<td>Print the specified ARGs to stdout</td>
</tr>
<tr>
<td>erase</td>
<td>Tool for erasing MTD partitions</td>
</tr>
<tr>
<td>eraseall</td>
<td>Tool for erasing entire MTD partitions</td>
</tr>
<tr>
<td>false*</td>
<td>Do nothing, unsuccessful</td>
</tr>
<tr>
<td>find</td>
<td>Search for files</td>
</tr>
<tr>
<td>flashw</td>
<td>Write data to individual flash devices</td>
</tr>
<tr>
<td>flatfsd</td>
<td>Daemon to save RAM file systems back to FLASH</td>
</tr>
<tr>
<td>ftp</td>
<td>Internet file transfer program</td>
</tr>
<tr>
<td>gen-keys</td>
<td>SSH key generation program</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>getopt</td>
<td>Parses command options</td>
</tr>
<tr>
<td>gettyd</td>
<td>Getty daemon</td>
</tr>
<tr>
<td>grep</td>
<td>Print lines matching a pattern</td>
</tr>
<tr>
<td>gunzip</td>
<td>Compress or expand files</td>
</tr>
<tr>
<td>gzip</td>
<td>Compress or expand files</td>
</tr>
<tr>
<td>hd</td>
<td>ASCII, decimal, hexadecimal, octal dump</td>
</tr>
<tr>
<td>hostname</td>
<td>Get or set hostname or DNS domain name</td>
</tr>
<tr>
<td>httpd</td>
<td>Listen for incoming HTTP requests</td>
</tr>
<tr>
<td>hwclock</td>
<td>Query and set hardware clock (RTC)</td>
</tr>
<tr>
<td>instd</td>
<td>Network super-server daemon</td>
</tr>
<tr>
<td>init</td>
<td>Process control initialization</td>
</tr>
<tr>
<td>ip</td>
<td>Show or manipulate routing, devices, policy routing and tunnels</td>
</tr>
<tr>
<td>ipmitool</td>
<td>Linux IPMI manager</td>
</tr>
<tr>
<td>iptables</td>
<td>Administration tool for IPv4 packet filtering and NAT</td>
</tr>
<tr>
<td>ip6tables</td>
<td>Administration tool for IPv6 packet filtering</td>
</tr>
<tr>
<td>iptables-restore</td>
<td>Restore IP Tables</td>
</tr>
<tr>
<td>ls</td>
<td>List directory contents</td>
</tr>
<tr>
<td>mail</td>
<td>Send and receive mail</td>
</tr>
<tr>
<td>mkdir</td>
<td>Make directories</td>
</tr>
<tr>
<td>mkfssfs</td>
<td>Create an MS-DOS file system under Linux</td>
</tr>
<tr>
<td>mknod</td>
<td>Make block or character special files</td>
</tr>
<tr>
<td>more</td>
<td>File perusal filter for crt viewing</td>
</tr>
<tr>
<td>mount</td>
<td>Mount a file system</td>
</tr>
<tr>
<td>msmtpl</td>
<td>SMTP mail client</td>
</tr>
<tr>
<td>mv</td>
<td>Move (rename) files</td>
</tr>
<tr>
<td>nc</td>
<td>TCP/IP Swiss army knife</td>
</tr>
<tr>
<td>netflash</td>
<td>Upgrade firmware on uCLinux platforms using the blkmem interface</td>
</tr>
<tr>
<td>netstat</td>
<td>Print network connections, routing tables, interface statistics etc</td>
</tr>
<tr>
<td>ntpd</td>
<td>Network Time Protocol (NTP) daemon</td>
</tr>
<tr>
<td>pggrep</td>
<td>Display process(es) selected by regex pattern</td>
</tr>
<tr>
<td>pidof</td>
<td>Find the process ID of a running program</td>
</tr>
<tr>
<td>ping</td>
<td>Send ICMP ECHO_REQUEST packets to network hosts</td>
</tr>
<tr>
<td>ping6</td>
<td>IPv6 ping</td>
</tr>
<tr>
<td>pkill</td>
<td>Sends a signal to process(es) selected by regex pattern</td>
</tr>
<tr>
<td>pmchat</td>
<td>Opengear command similar to the standard chat command (via portmanager)</td>
</tr>
<tr>
<td>pmdeny</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pmind</td>
<td>Opengear command similar to the standard tip or cu but all serial port access is directed via the portmanager.</td>
</tr>
<tr>
<td>pmloggerd</td>
<td>Opengear command to query portmanager for active user sessions</td>
</tr>
<tr>
<td>pmshell</td>
<td>Opengear command that handles all serial port access</td>
</tr>
<tr>
<td>portmap</td>
<td>DARPA port to RPC program number mapper</td>
</tr>
<tr>
<td>pppd</td>
<td>Point-to-Point protocol daemon</td>
</tr>
<tr>
<td>ps *</td>
<td>Report a snapshot of the current processes</td>
</tr>
<tr>
<td>pwd *</td>
<td>Print name of current/working directory</td>
</tr>
<tr>
<td>reboot *</td>
<td>Soft reboot</td>
</tr>
<tr>
<td>rm *</td>
<td>Remove files or directories</td>
</tr>
<tr>
<td>rmdir *</td>
<td>Remove empty directories</td>
</tr>
<tr>
<td>routed</td>
<td>Show or manipulate the IP routing table</td>
</tr>
<tr>
<td>routed</td>
<td>Show or manipulate the IP routing table</td>
</tr>
<tr>
<td>routef</td>
<td>IP Route tool to flush IPv4 routes</td>
</tr>
<tr>
<td>routel</td>
<td>IP Route tool to list routes</td>
</tr>
<tr>
<td>rtacct</td>
<td>Applet printing /proc/net/rt_acct</td>
</tr>
<tr>
<td>rmon</td>
<td>RTnetlink listener</td>
</tr>
<tr>
<td>scp</td>
<td>Secure copy (remote file copy program)</td>
</tr>
<tr>
<td>sed *</td>
<td>Text stream editor</td>
</tr>
<tr>
<td>setmac</td>
<td>Sets the MAC address</td>
</tr>
<tr>
<td>setserial</td>
<td>Sets and reports serial port configuration</td>
</tr>
<tr>
<td>sh</td>
<td>Shell</td>
</tr>
<tr>
<td>showmac</td>
<td>Shows MAC address</td>
</tr>
<tr>
<td>sleep *</td>
<td>Delay for a specified amount of time</td>
</tr>
<tr>
<td>smbmnt</td>
<td>Helper utility for mounting SMB file systems</td>
</tr>
<tr>
<td>smbmount</td>
<td>Mount an SMBFS file system</td>
</tr>
<tr>
<td>smbmount</td>
<td>SMBFS umount for normal users</td>
</tr>
<tr>
<td>smnd</td>
<td>SNMP daemon</td>
</tr>
<tr>
<td>snmptrap</td>
<td>Sends an SNMP notification to a manager</td>
</tr>
<tr>
<td>sredir</td>
<td>RFC 2217 compliant serial port redirector</td>
</tr>
<tr>
<td>ssh</td>
<td>OpenSSH SSH client (remote login program)</td>
</tr>
<tr>
<td>ssh-keygen</td>
<td>Authentication key generation, management, and conversion</td>
</tr>
<tr>
<td>sshd</td>
<td>OpenSSH SSH daemon</td>
</tr>
<tr>
<td>sslwrap</td>
<td>Program that allows plain services to be accessed via SSL</td>
</tr>
<tr>
<td>stty</td>
<td>Change and print terminal line settings</td>
</tr>
<tr>
<td>stunnel</td>
<td>Universal SSL tunnel</td>
</tr>
<tr>
<td>sync *</td>
<td>Flush file system buffers</td>
</tr>
<tr>
<td>sysctl</td>
<td>Configure kernel parameters at runtime</td>
</tr>
<tr>
<td>syslogd</td>
<td>System logging utility</td>
</tr>
<tr>
<td>tar *</td>
<td>The tar archiving utility</td>
</tr>
<tr>
<td>tc</td>
<td>Show traffic control settings</td>
</tr>
<tr>
<td>tcpdump</td>
<td>Dump traffic on a network</td>
</tr>
<tr>
<td>telnetd</td>
<td>Telnet protocol server</td>
</tr>
<tr>
<td>tftp</td>
<td>Client to transfer a file from/to tftp server</td>
</tr>
<tr>
<td>tftp</td>
<td>Trivial file Transfer Protocol (tftp) server</td>
</tr>
<tr>
<td>tip</td>
<td>Simple terminal emulator/cu program for connecting to modems and serial devices</td>
</tr>
<tr>
<td>top</td>
<td>Provide a view of process activity in real time</td>
</tr>
</tbody>
</table>
### Commands

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

Commands above which are appended with ‘*’ come from Busybox (the Swiss Army Knife of embedded Linux) at [http://www.busybox.net/downloads/BusyBox.html](http://www.busybox.net/downloads/BusyBox.html).

Others are generic Linux commands and most commands take the `-h` or `--help` argument to provide a terse runtime description of their behavior. More details on the generic Linux commands can be found online at [http://en.tldp.org/HOWTO/HOWTO-INDEX/howtos.html](http://en.tldp.org/HOWTO/HOWTO-INDEX/howtos.html) and [http://www.faqs.org/docs/Linux-HOWTO/Remote-Serial-Console-HOWTO.html](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](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](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](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](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](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.
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] ... ]
bg [job_spec]
bind [-ipvsPVS] [-m keymap] [-f f i]
break [n]
builtin [shell-builtin [arg ...]]
case WORD in [PATTERN [...] PATTERN]
  cd [-PL] [dir]
command [-pVv]
command [arg ...]
compgen [-abcdefjkvu] [-o option]
complete [-abcdefjkvu] [-pr] [-o o]
continue [n]
declare [-afFrxi] [-p] name[=value]
dirs [-clpv] [+N] [-N]
disown [-h] [-ar] [jobspec ...]
echo [-neE] [arg ...]
enable [-pnds] [-a] [-f filename]
eval [arg ...]
exec [-cl] [-a name] file [redirect]
ext [n]
export [-nf] [name ...] or export false
fc [-e ename] [-nlr] [first] [last]
fg [job_spec]
local name[=value] ...
logout
popd [-N] [-N] [-n]
printf format [arguments]
pushd [dir] [+N] [-N] [-n]
pwd [-PL]
read [-ers] [-t timeout] [-p promp]
readonly [-anf] [name ...] or read return [n]
select NAME [in WORDS ...:] do COMMANDS
set [-abefknptuvxBCHP] [-o opti]
shift [n]
shopt [-pqsu] [-o long-option] opt
source filename
suspend [-f]
test [expr]
time [-p] PIPELINE
times
trap [arg] [signal_spec ...]
true
type [-apt] name [name ...]
typeset [-afRxil] [-p] name[=value]
ulimit [-SHcdflmnpstuv] [limit]
umask [-p] [-S] [mode]
```
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 [-Inprs] [jobspec ...] or job kill
[-s sigspec / -n signum / -si let arg [arg
...]
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 ; }
## APPENDIX B: TERMINOLOGY

<table>
<thead>
<tr>
<th>TERM</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>3G</td>
<td>Third-generation cellular technology. The standards that determine 3G call for greater bandwidth and higher speeds for cellular networks</td>
</tr>
<tr>
<td>AES</td>
<td>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.</td>
</tr>
<tr>
<td>APN</td>
<td>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</td>
</tr>
<tr>
<td>Authentication</td>
<td>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</td>
</tr>
<tr>
<td>BIOS</td>
<td>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</td>
</tr>
<tr>
<td>Bonding</td>
<td>Ethernet Bonding or Failover is the ability to detect communication failure transparently, and switch from one LAN connection to another.</td>
</tr>
<tr>
<td>BOOTP</td>
<td>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</td>
</tr>
<tr>
<td>Certificates</td>
<td>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.</td>
</tr>
<tr>
<td>Certificate Authority</td>
<td>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.</td>
</tr>
<tr>
<td>Certificate Revocation List</td>
<td>A list of certificates that have been revoked by the CA before they expired. This may be necessary if the private key certificate has been compromised or if the holder of the certificate is to be denied the ability to establish a connection to the console server.</td>
</tr>
<tr>
<td>CHAP</td>
<td>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.</td>
</tr>
<tr>
<td>CMS</td>
<td>The term CMS refers to the Centralized Management Software running in all the Lighthouse appliances</td>
</tr>
<tr>
<td>Console server</td>
<td>The term console server refers generically to the Opengear datacenter and remote management appliances, including the ACM5000, ACM5500, IM4200, CM41000 and SD4000 product lines.</td>
</tr>
<tr>
<td>DES</td>
<td>The Data Encryption Standard is a block cipher with 64-bit blocks and a 56-bit key.</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol. A communications protocol that assigns IP addresses to computers when they are connected to the network.</td>
</tr>
<tr>
<td><strong>DNS</strong></td>
<td>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.</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>DUN</strong></td>
<td>Dial Up Networking</td>
</tr>
<tr>
<td><strong>Encryption</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Ethernet</strong></td>
<td>A physical layer protocol based upon IEEE standards</td>
</tr>
<tr>
<td><strong>Firewall</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Gateway</strong></td>
<td>A machine that provides a route (or pathway) to the outside world.</td>
</tr>
<tr>
<td><strong>Hub</strong></td>
<td>A network device that allows more than one computer to be connected as a LAN, usually using UTP cabling.</td>
</tr>
<tr>
<td><strong>Internet</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Intranet</strong></td>
<td>A private TCP/IP network within an enterprise.</td>
</tr>
<tr>
<td><strong>IPMI</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Key lifetimes</strong></td>
<td>The length of time before keys are renegotiated</td>
</tr>
<tr>
<td><strong>LAN</strong></td>
<td>Local Area Network</td>
</tr>
<tr>
<td><strong>LDAP</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>LED</strong></td>
<td>Light-Emitting Diode</td>
</tr>
<tr>
<td><strong>Lighthouse appliance</strong></td>
<td>This term refers generically to the Lighthouse VM software appliance, the Lighthouse Standard hardware appliance and Lighthouse Enterprise hardware appliance</td>
</tr>
<tr>
<td><strong>MAC address</strong></td>
<td>Every piece of Ethernet hardware has a unique number assigned to it called its MAC address. Ethernet is used locally to connect the console server to the Internet, and it may share the local network with many other appliances. The MAC address is used by the local Internet router in order to direct console server traffic to it rather than 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 console server has a MAC address listed on a label underneath the device.</td>
</tr>
<tr>
<td><strong>Managed Console Server</strong></td>
<td>Managed Console Server refers generically to any console server that is being centrally managed by a Lighthouse appliance.</td>
</tr>
<tr>
<td><strong>MSCHAP</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Terminology</strong></td>
<td><strong>Definition</strong></td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
</tr>
<tr>
<td><strong>NAT</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Net mask</strong></td>
<td>The way that computers know which part of a TCP/IP address refers to the network, and which part refers to the host range.</td>
</tr>
<tr>
<td><strong>NFS</strong></td>
<td>Network File System is a protocol that allows file sharing across a network. Users can view, store, and update files on a remote computer.</td>
</tr>
<tr>
<td><strong>NTP</strong></td>
<td>Network Time Protocol (NTP) used to synchronize clock times in a network of computers.</td>
</tr>
<tr>
<td><strong>OUT OF BAND</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>PAP</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>PPP</strong></td>
<td>Point-to-Point Protocol. A networking protocol for establishing simple links between two peers.</td>
</tr>
<tr>
<td><strong>RADIUS</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Router</strong></td>
<td>A network device that moves packets of data. A router differs from hubs and switches because it is &quot;intelligent&quot; and can route packets to their final destination.</td>
</tr>
<tr>
<td><strong>SIM</strong></td>
<td>Subscriber Identity Module (SIM) card stores unique serial numbers and security authentication used to identify a subscriber on mobile telephony devices.</td>
</tr>
<tr>
<td><strong>SMASH</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>SMTP</strong></td>
<td>Simple Mail Transfer Protocol. console server includes, SMTPclient, a minimal SMTP client that takes an email message body and passes it on to a SMTP server (default is the MTA on the local host).</td>
</tr>
<tr>
<td><strong>SOL</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>SSH</strong></td>
<td>Secure Shell is secure transport protocol based on public-key cryptography.</td>
</tr>
<tr>
<td><strong>SSL</strong></td>
<td>Secure Sockets Layer is a protocol that provides authentication and encryption services between a web server and a web browser.</td>
</tr>
<tr>
<td><strong>TACACS+</strong></td>
<td>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...</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>TCP/IP</th>
<th>Transmission Control Protocol/Internet Protocol. The basic protocol for Internet communication.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/IP address</td>
<td>Fundamental Internet addressing method that uses the form nnn.nnn.nnn.nnn.</td>
</tr>
<tr>
<td>Telnet</td>
<td>Telnet is a terminal protocol that provides an easy-to-use method of creating terminal connections to a network.</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UTC</td>
<td>Coordinated Universal Time.</td>
</tr>
<tr>
<td>UTP</td>
<td>Unshielded Twisted Pair cabling. A type of Ethernet cable that can operate up to 100Mb/s. Also known as Category 5 or CAT 5.</td>
</tr>
<tr>
<td>VNC</td>
<td>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.</td>
</tr>
<tr>
<td>VPN</td>
<td>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</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
</tr>
<tr>
<td>WINS</td>
<td>Windows Internet Naming Service (WINS) that manages the association of workstation names and locations with IP addresses</td>
</tr>
</tbody>
</table>