Avocent®

CYCLADES® ACS 5000

Command Reference Guide

FCC Warning Statement

The Cyclades ACS 5000 advanced console server has been tested and found to comply with the limits for Class A digital devices, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment.

This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the Installation and Service Manual, may cause harmful interference to radio communications.

Operation of this equipment in a residential area is likely to cause harmful interference in which case the user is required to correct the problem at his or her own expense.

Notice about FCC Compliance for All Cyclades ACS 5000 Advanced Console Server Models

To comply with FCC standards, the Cyclades ACS 5000 advanced console server requires the use of a shielded CAT 5 cable for all interface ports. Notice that this cable is not supplied with either of the products and must be provided by the customer.

Canadian DOC Notice

The Cyclades ACS 5000 advanced console server does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

L'Cyclades ACS 5000 advanced console server n'émete pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe A prescrites dans le règlement sur le brouillage radioélectrique edicté par le Ministère des Communications du Canada.

Cyclades® ACS 5000 Advanced Console Server

Command Reference Guide

Instructions

This symbol is intended to alert the user to the presence of important operating and maintenance (servicing) instructions in the literature accompanying the appliance.

Dangerous Voltage

This symbol is intended to alert the user to the presence of uninsulated dangerous voltage within the product's enclosure that may be of sufficient magnitude to constitute a risk of electric shock to persons.

Power On

This symbol indicates the principal on/off switch is in the on position.

Power Off

This symbol indicates the principal on/off switch is in the off position.

Protective Grounding Terminal

This symbol indicates a terminal which must be connected to earth ground prior to making any other connections to the equipment.

TABLE OF CONTENTS

CHAPTER 1

Using the Command Line Interface

Overview

The Cyclades® ACS 5000 advanced console server command line interface (CLI) may be used for administration and maintenance of the ACS 5000 console server. CLI is comprised of a set of keywords nested in a hierarchy format. CLI allows the console server administrator to perform the same configuration tasks available through the web manager. In addition, it allows executing the frequently performed configuration tasks saved in text files in batch mode or through shell scripts.

Understanding the CLI Utility

The CLI utility is built on a set of commands that are nested in a hierarchical format. Some commands require parameters that are user-defined.

For example, network configuration tasks include network, hostsetting and hostname commands nested in the following format.

```
cli> config network hostsettings hostname [parameter]
```

Commands used to configure or change a set of parameters:

```
cli> config security adduser username john password john12 admin yes
shell /bin/sh
```

Commands may also specify a function or an action to be performed. For example,

```
cli> config runconfig
cli> config savetoflash
```

Accessing the CLI

The CLI may be accessed in any of the following three methods:

- By local logins through the console port

  Local console server “root” users may access the command line by logging in through the console port using a terminal or a server running a terminal emulation program.

- By remote logins using SSH, PPP or a terminal emulation program

Remote users may access the console server CLI through SSH, by using a terminal emulation program to dial into an external modem or by creating a PPP connection with an external modem.

- By clicking *Connect to ACS 5000* in the web manager.

  After logging into the web manager, you may access the CLI by clicking the *Connect* menu option.

Important features of the CLI utility

- Only one user logged in as “root” or “admin” may have an active CLI or web manager session. A second user who connects through the CLI or the web manager as “root” or “admin” has a choice to abort the session or close the other user’s session.

NOTE: If there are cron jobs running through automated scripts, a root or admin user login may cause the automated cron jobs to fail.

- CLI has three possible user levels:
  - Root user - A Linux root user has access to the full functionality of the CLI interface. Root users have access to the shell command in the CLI that provides access to the console server shell prompt.

NOTE: An administrator may enforce the Linux shell to execute the CLI utility when the user logs into the console server (/bin/CLI). A user with “root” access may invoke the Linux shell from the CLI interface. An admin or a regular user who is configured with CLI as the default shell may not access the Linux shell.

  - Admin - A Linux admin user has access to the full functionality of the CLI except the shell command, which provides access to the console server Linux shell prompt.
  - Regular user - A Linux regular user has access only to limited functionality of the CLI. Access is granted only to the applications commands of the CLI utility.
- CLI interface generates syslog messages for executed commands, and when sessions are terminated. For example,

```
Apr 19 17:51:44 src_dev_log@swes-129 CLI[413]: User root starts an
interactive CLI session.cli>config

Apr 19 16:28:02 src_dev_log@swes-129 CLI[412]: Session closed due
idletimeout

Apr 19 17:54:23 src_dev_log@swes-129 CLI[413]: User root executed
[quit]
```

- CLI writes every command executed in interactive mode in the file ~/.history. This file stores the last 1000 commands executed in any CLI session.

Modes of operation

The following table describes the three modes of executing commands using the CLI utility.

Table 1.1: Modes of Operation

Mode	Description
Command Line	CLI is invoked in the Linux shell with commands and parameters. For example: `[root@CAS root]# bin/CLI config network hostsettings hostname <parameter>`

Batch	• CLI commands may be saved in a text file and executed in batch mode by invoking the CLI utility with the -f <filename> option.
	• CLI commands may be used in a shell script. For example, #/bin/CLI may be invoked at the top of a shell script if the script contains only CLI commands. Any type of shell may be used to run CLI commands along with other commands.
	
	For example:
	
	• Create a script that calls /bin/CLI to configure a hostname in batch mode.
	`#!/bin/CLI`
	`config network hostsettings hostname FremontACS`
	`config savetoflash`
	`:wq`
	
	• Run a CLI command from the same script that is running other Linux commands.
	#!/bin/bash
	...
	`/bin/CLI -s config network hostsettings hostname FremontACS`
	...
	
	• Run multiple CLI commands from a script that is running other Linux commands.
	`#!/bin/bash`
	`...`
	`/bin/CLI << EOF`
	`config network hostsettings hostname FremontACS`
	`config security adduser username johndoe`
	`config savetoflash`
	`EOF`

Table 1.1: Modes of Operation (Continued)

Mode	Description

Interactive	CLI is invoked and commands and parameters are entered in the Linux shell. CLI is active until the **quit** command is issued. For example,
	`CLI> config network hostsettings dhcp <yes>`
	`CLI> config runconfig`
	`CLI> config savetoflash`
	`CLI> config quit`
	`[root@CAS root]#`

CLI Navigation

Autocompletion

Autocompletion may be used to find out what commands and parameters are available.

- Pressing the **Tab** key twice displays all the commands at the top level. For example:

  `cli>` **Tab Tab**

  ```
  administration    info         return   version
  applications      portStatus   shell
  config            quit         show
  ```

- Pressing the **Tab** key once after partially-typing a command automatically completes the parameter name. If there is more than one parameter name beginning with the typed characters, then pressing the **Tab** key again displays them all. For example:

  `cli>` **`i Tab`**

  `info`

  `cli>` **`a Tab Tab`**

  ```
  administration applications
  ```

- Pressing the **Tab** key after the first level command displays the commands one level down in the hierarchy. For example:

  `cli>` **`config Tab`**

  ```
  administration  discardchanges   physicalports     savetoflash
  applications    ipmi             restorefromflash  security
  autodiscovery   network          runconfig         virtualports
  ```

Saving CLI changes

Configuration changes made in any of the CLI modes are temporary. Changes are not activated and saved into the configuration files unless you run the commands described in the following table.

Table 1.2: CLI Commands for Saving Configuration Changes

Command	Action
config runconfig	Saves and activates configuration changes in the appropriate configuration files.
config savetoflash	Saves any unsaved configuration changes in the configuration files and creates a zipped backup copy of the files in a backup directory for possible later retrieval.
config discardchanges	Restores the backed up configuration files, overwriting any configuration changes made since the last time the savetoflash option was executed.

Using CLI hotkeys

The CLI hotkeys may be used to perform the following types of actions:

- Move the cursor on the command line.
- Move through the list of commands in the command history.
- Edit characters on the command line.

Table 1.3: Cursor Movement Keys

Keyboard Keys	Description
Ctrl+a	Move to the start of the current line.
Ctrl+e	Move to the end of the line.
Ctrl+b	Move back a character (same as the left arrow key).
Ctrl+f	Move forward a character (same as the right arrow key).
Esc+b	Move back to the start of the current or previous word. Words are composed of letters and digits.
Esc+f	Move forward to the end of the next word. Words are composed of letters and digits.
Ctrl+l	Clear the screen and redraw the current line, leaving the current line at the top of the screen.

Table 1.4: Command History Keys

Keyboard Keys	Description
Ctrl+n	Move forward through the history list, fetching the next command (same as <down arrow key>).

Table 1.4: Command History Keys (Continued)

Ctrl+p	Move back through the history list, fetching the previous command (same as <up arrow key>).

NOTE: The command history buffer is only available for the last 500 commands in the current session. The history is cumulative, so terminating the session does not clear the buffer. This means a user may log in to the CLI and go back over the commands entered by a previous user.

Table 1.5: Text Modification Keys

Keyboard Keys	Description
Ctrl+d	Delete the character under the cursor (same as **Delete** key).
Ctrl+h	Same as **Backspace** key.
Ctrl+k	Clear the text from the cursor to the end of the line.
Ctrl+u	Clear backward from the cursor to the beginning of the current line.
Ctrl+w	Delete the word behind point.
Esc+d	Clear from the cursor to the end of the current word, or if between words, to the end of the next word.

Esc+Tab	Displays the current parameter of the command entered. You may edit the value. For example: To display the current value for domain and edit it.
	`cli> config network hostsettings`
	`hostsettings> domain [press <Esc> <Tab>]`
	`hostsettings> domain avocent.com`

CLI Global commands

The CLI global commands may be entered at any level of the CLI hierarchy.

Table 1.6: CLI Global Commands

Command	Description
quit	Ends the CLI session.
return	Goes up one level in the CLI hierarchy.
info	Displays the help information available for the current level in the hierarchy. When combined with a command name supported at the current level, the applicable information or parameter is displayed.

Table 1.6: CLI Global Commands (Continued)

show	Displays the configuration parameter(s). When combined with a command name supported at the current level, the applicable information or parameter is displayed.

CLI command arguments

Command arguments are used when CLI is invoked in the command line mode in the Linux shell or in a batch mode

Table 1.7: CLI Command Arguments

Argument	Description
-q	Suppress the output of error messages from CLI.
-t <time>	Timeout in minutes, default is 10 minutes.
-T	Disable the idle time-out. Same as -t 0.
-s batch mode only	Save changes to flash. This is the same as savetoflash command.
-r batch mode only	Activate changes. This is the same as runconfig command.
-f <filename>	Executes the commands in the file <filename>.

CHAPTER 2

Network Configuration

Network Settings

The following instructions assume you are installing a new console server or you have reset an existing unit to factory default parameters.

Default configuration is with IPV4 and IPV6 enabled:

- IPV4 networking will be enabled and the main Ethernet interface IP address will be obtained from a DHCPv4 Server.
- IPV6 networking will be enabled only for the basic services of the main Ethernet interface and its IPv6 address will be obtained from a local router (stateless only option).

To configure initial network parameters using the wiz command:

1. From your terminal emulation application, log in to the console port as **root**. The default password is **avocent**.

NOTE: It is strongly recommended to change the default password to a new password before configuring the console server for secure access.

2. To change a password, run the following command.

   ```
   [root@CAS root]# passwd
   New password:
   ```

3. Launch the configuration wizard by entering the wiz command.

   ```
   [root@CAS root]# wiz
   ```

4. The system displays a configuration wizard banner, instructions for using the utility and the current configuration.
5. At the prompt, Set to defaults?, enter **n** to change the defaults.
6. Continue through the configuration parameters until you are prompted to determine if the parameters are correct.

   ```
   Are all these parameters correct? (y/n) [n] :
   ```

7. Enter **n** to go back and change any configuration parameters.

-or-

If you enter **y**, you will be prompted to save your configuration after the following warning is displayed:

```
(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)

Do you want to activate your configurations now? (y/n) [y] :
```

8. Activate and save your configuration when prompted to do so.
9. To confirm the configuration, enter the **ifconfig** command. The new network settings will be displayed.

IPv4 and IPv6 addressing

NOTE: All of the following configuration parameters are available in the wizard (wiz).

CAUTION: If you are accessing the CLI through a network connection instead of the through a console port, you risk losing network access and control of the console server when you change the IP mode or the IP address. Be sure to keep track of the new IP address before activating the new configuration, so you can reconnect.

By default, IPv4 and IPv6 network addressing will be enabled. The console server allows the following network addressing configurations:

- IPv4 only
- IPv6 only
- Dual Stack (IPv4 and IPv6)

Disabling IPv4

If you disable IPv4, configuration of IPv4 addresses will not be allowed. A warning message will display advising you that services not supporting IPv6 will be unavailable. The IPv4 tab will be disabled.

Disabling/Enabling IPv6

If you disable IPv6, configuration of IPv6 addresses will not be allowed and the IPv6 tab will be disabled. If you change IPv6 from disabled to enabled, a warning message will display advising you that some services not supporting IPv6 will be unavailable. You will have to configure those services supporting IPv6 for proper operation.

NOTE: If services not supporting IPv6 are needed, select Dual Stack (IPv4 and IPv6) and those services will be available for IPv4.

IPv4 Addressing

To enable IPv4 network addressing:

1. From the shell prompt on your terminal, enter the following command.
   # **CLI**
2. From the cli prompt, enter the following:
   cli> **config network hostsettings ipmode dualstack**

   This will enable both IPv4 and IPv6 network addressing.

   -or-

   cli> **config network hostsettings ipmode ipv4**

   This will enable IPv4 network addressing only.

To set IPv4 specific configurations:

From the cli prompt, enter the following.

cli> **config network hostsettings**

Follow the parameters in Table 2.1 for the rest of the configuration.

Table 2.1: IPv4 Specific Configurations

IPv4 Specific Level 1	IPv4 Specific Level 2	Description
primipaddress	<nnn.nnn.nnn.nnn>	The primary IP address of the console server - automatically obtained if DHCP is enabled
secipaddress	<nnn.nnn.nnn.nnn>	The secondary IP address of the console server
primsubnetmask	<nnn.nnn.nnn.nnn>	Subnet mask for the primary IP address
secsubnetmask	<nnn.nnn.nnn.nnn>	Subnet mask for the secondary IP address
dhcp	<nnn.nnn.nnn.nnn>	An IPv4 address will be dynamically obtained from a DHCPv4 server

IPv6 addressing

Services not supported in IPv6

IPv6 does not support the following services:

- NIS authentication
- NFS data logging

- Virtual ports

To enable IPv6 network addressing:

1. From the shell prompt on your terminal, enter the following command.
   # **CLI**
2. From the cli prompt, enter the following:
   cli> **config network hostsettings ipmode dualstack**

   This will enable both IPv4 and IPv6 network addressing.

   -or-

   cli> **config network hostsettings ipmode ipv6**

   This will enable IPv6 network addressing only.

To set IPv6 specific configurations:

From the cli prompt, enter the following:

cli> **config network hostsetings ipv6**

Follow the parameters in Table 2.2 for the rest of the configuration.

Table 2.2: IPv6 Specific Configurations

IPv6 Specific Level 1	IPv6 Specific Level 2	Description
dhcp6		Selects the options for the information that will be retrieved from the DHCPv6 server.
	none	No further data will be retrieved from the server.
	dns	The DNS server IP address will be retrieved from the server.
	domain	The domain path will be retrieved from the server.
	dns_domain	The DNS server IP address and the domain path will be retrieved from the server.

Table 2.2: IPv6 Specific Configurations (Continued)

IPv6 Specific Level 1	IPv6 Specific Level 2	Description
ipv6method		Selects the way IPV6 addresses will be configured or obtained.
	stateless_only	IPv6 local address will be dynamically obtained from an IPv6 Router in the local network – this method is to be used only if the two others are not available (local IPv6 addresses obtained by the router cannot be used outside the local network).
	static	IPv6 address will be statically configured.
	dhcp	IPv6 address and its prefix length will be dynamically obtained from a DHCPv6 server.
staticipaddress	<ipaddress>/<prefix_length>	Configures a static IPv6 address and its prefix length for the interface. This is available only if ipv6method is configured as *static*.

To configure a static primary IP address in IPv6 mode, enter the following:

```
cli> config network hostsettings ipv6 staticpaddress <IPv6_address>
```

To configure a dynamic primary IP address in IPv6 mode, enter the following:

```
cli> config network hostsettings ipv6 ipv6method stateless_only
```

-or-

```
cli> config network hostsettings ipv6 ipv6method dhcp
```

IPv4 and IPv6 common parameters

To set up parameters common to IPv4 and IPv6 mode:

To set up or change the primary DNS server, enter the following:

```
cli> config network hostsettings primdnsserver <primary_DNS_server_ip>
```

Similarly, configure the secondary DNS server, if necessary:

```
cli> config network hostsettings secdnsserver <secondary_DNS_server_ip>
```

To set up or change the domain name where your system resides, enter the following:

```
cli> config network hostsettings domain <domain_name>
```

To configure the gateway, enter the following:

```
cli> config network stroutes add default gateway <gateway_IP_address>
```

NOTE: If the gateway address is IPV6 link_local (range identified by the first 10 bits equal to 1111111010), then the interface id is required: **config network stroutes add default gateway <gateway_IP_address> interface <interface_ID>**

Activate and save your configuration.

```
cli> config runconfig
cli> config savetoflash
```

Host settings

To configure host settings:

1. Enter the following string at the CLI prompt. Refer to Table 2.3 for host settings parameters and values.

   ```
   cli> config network hostsettings <parameter> <value>
   ```

2. Activate and save your configuration.

Table 2.3: Host Settings Parameters and Values

Parameter Level1	Parameter Level2	Value	Description
banner		<console banner>	Banner for the user shell
bonding			Redundancy for the ethernet interface
	miimon	<number>	The interval in which the active interface is checked to see if it is still communicating (in milliseconds)
	updelay	<number>	The time the system waits to make the primary interface active after it has been detected as up (in milliseconds)
dhcp		yes|no	Enable or disable DHCP
domain		<domain name>	Domain name
hostname		<string>	Console Server name
mtu		<number[200-1500]>	Maximum Transmission Unit used by the TCP protocol
primdnsserver		<IPv6_address>/<prefix_length>	Primary DNS Server (IPv4 or IPv6)

Table 2.3: Host Settings Parameters and Values (Continued)

Parameter Level1	Parameter Level2	Value	Description
secdnsserver		<IPv6_address>/<prefix_length>	Secondary DNS Server (IPv4 or IPv6)
primipaddress		<nnn.nnn.nnn.nnn>	Primary IP address (IPv4 specific)
secipaddress		<nnn.nnn.nnn.nnn>	Secondary IP address (IPv4 specific)
primsubnetmask		<nnn.nnn.nnn.nnn>	Primary subnet mask (IPv4 specific)
secsubnetmask		<nnn.nnn.nnn.nnn>	Secondary subnet mask (IPv4 specific)

Security Profiles

A security advisory appears the first time the console server is turned on, or when the unit is reset to factory default parameters. Once you have configured the basic network settings, a security profile must be selected in order to proceed to further configuration procedures. Table 2.4 describes the protocols and services available for each security profile.

Table 2.4: Security Profiles

Security profile	Description
Secured	Predefined security profile. All protocols and services are disabled except SSHv2, HTTPs and SSH to Serial Ports.
Moderate (Default)	Predefined security profile. Enables SSHv1, SSHv2, HTTP, HTTPs, Telnet, SSH and Raw connections to serial ports, ICMP and HTTP redirection to HTTPs.
Open	Predefined security profile. Enables all services, Telnet, SSHv1, SSHv2, HTTP, HTTPS, SNMP, RPC, ICMP and Telnet, SSH and Raw connections to Serial Ports.
Custom	Administrator may configure individual protocols and services and configure access to serial ports.

To select a predefined security profile:

Configure a predefined security profile by entering the following string at the CLI prompt.

```
cli> config security profile [secured|moderate|open]
```

To configure a custom security profile:

1. Navigate to the custom menu.

```
cli> config security profile custom
```

2. Enable or disable desired protocols or services. Refer to Table 2.5 for the list of parameters and values.

   ```
   custom> [parameter] <value>
   ```

3. Activate and save your configuration.

Table 2.5: Custom Security Profile Parameters

Parameter Level1	Parameter Level2	Parameter Level3	Value
ftp			yes|no
icmp			yes|no
ipsec			yes|no
ports>	auth2		yes|no
	bidirect		yes|no
	raw2sport		yes|no
	ssh2sport		yes|no
	telnet2sport		yes|no
rpc			yes|no
snmp			yes|no
ssh>	root_access		yes|no
	ssh_x509>	CA_file	
		hostkey	<path and filename of CA certificate>
		authorizedkeys	<path and filename of authorized keys>
	sshd_port		<number>
	sshv1		yes|no
	sshv2		yes|no
telnet			yes|no
web>	http		yes|no
	http2https		yes|no
	http_port		<number>
	https		yes|no
	https_port		<number>

Enable serial ports

By default, the console server is configured with all serial ports disabled.

To enable serial ports:

1. Enable single or multiple serial ports.

   ```
   cli> config physicalports <range/list[1-32]> enable yes
   ```

2. Activate and save your configuration.

VPN Configuration

You can set up VPN connections to establish an encrypted communication between the console server and a host on a remote network. The encryption creates a security tunnel for dedicated communications.

To set up a security gateway, you should install IPSec. The ESP and AH authentication protocols, and RSA Public Keys and Shared Secret are supported.

To configure VPN:

1. Execute the following command to enable IPSec.

   ```
   cli> config security profile custom ipsec <yes>
   ```

2. Configure VPN parameters, see Table 2.6.

   ```
   cli> config network vpn [parameter] <value>
   ```

3. Activate and save your configuration.

Table 2.6: VPN Parameters

Parameter	Value	Description
add	<connection name>	A name to identify the connection.
authmethod	<rsapubkey|sharesecret>	Authentication method used. Either RSA Public Key or Shared Secret.
authprotocol	<ah|esp>	Authentication protocol used. Either Encapsulating Security Payload (ESP) or Authentication Header (AH).
bootaction	<add|ignore|start>	The boot action configured for the host.
leftid rightid	hostname@xyz.com	This is the hostname that a local system and a remote system use for IPSec negotiation and authentication. It may be a fully qualified domain name (FQDN) preceded by @. For example, hostname@xyz.com.
leftip rightip	<IP_address>	The IPv4 or IPv6 address of the host.
leftnexthop rightnexthop	<IP_address>	The IPv4 or IPv6 address of the router through which the console server (on the left side) or the remote host (on the right side) sends packets to a host on a network.
leftrsakey rightrsakey	<string>	You need to generate a public key for the console server and find out the key used on the remote gateway. You may use copy and paste to enter the key in the RSA Key field.

Table 2.6: VPN Parameters (Continued)

Parameter	Value	Description
leftsubnet rightsubnet	<n.n.n.n/n>	The netmask of the subnetwork where the host resides. **NOTE:** Use CIDR notation. The IP number followed by a slash and the number of 'one' bits in the binary notation of the netmask. For example, 192.168.0.0/24 indicates an IP address where the first 24 bits are used as the network address. This is the same as 255.255.255.0.
secret	<string>	Pre-shared password between left and right users.

SNMP

Simple Network Management Protocol (SNMP) works by sending messages called protocol data units (PDUs) to different parts of a network. SNMP compliant devices, called agents, store data about themselves in Management Information Bases (MIBs) and return this data to the SNMP requesters. The console server uses the net-snmp package. See http://www.net-snmp.org for more information.

NOTE: Check the SNMP configuration before gathering information about the console server by SNMP. There are different types of attacks an unauthorized user may implement to retrieve sensitive information contained in the MIB. By default, the SNMP configuration in the console server does not permit the public community to read SNMP information.

In order to configure SNMP v1/v2, enter the following command. Refer to Table 2.7 for a list of parameters.

```
cli> config network snmp v1/v2 [parameter] <value>
```

Table 2.7: SNMP v1/v2 Configuration Parameters

Parameter	Value	Description
syscontact	<string>	The email address of the console server administrator.
syslocation	<string>	The physical location of the console server.
community	<string>	The group to which devices and management stations running SNMP belong.
oid	<string>	Object Identifier. Each managed object has a unique identifier.
permission	<string>	Read Only access to the entire Management Information Base (MIB) except for SNMP configuration objects. Read/Write access to the entire MIB except for SNMP configuration objects.
source	<string>	The host IP address.

To configure SNMP v1/v2 (example):

1. The following command configures SNMP v1/v2 with the following parameters.
   - community: avocent
   - OID: .1
   - permission: ro (read only)
   - source (allowed host): 192.168.0.200

   ```
   cli> config network snmp v1v2 add community avocent oid .1 permission
   ro source 192.168.0.200
   ```

2. Run the following commands to activate and save the configuration.

In order to configure SNMP v3, enter the following command. Refer to Table 2.8 for a list of parameters.

```
cli> config network snmp v3 [parameter] <value>
```

Table 2.8: SNMP v3 Parameters

Parameter	Value	Description
syscontact	<string>	The email address of the console server administrator.
syslocation	<string>	The physical location of the console server.
oid	<string>	Object Identifier. Each managed object has a unique identifier.
password	<string>	User password.
permission	<string>	Read Only access to the entire Management Information Base (MIB) except for SNMP configuration objects. Read/Write access to the entire MIB except for SNMP configuration objects.
username	<string>	Username.

To configure SNMP v3 (example):

1. The following command configures SNMP v3 with the following parameters.
   - username: john
   - password: john1234
   - OID: .1
   - permission: ro (read only)

NOTE: The SNMP v3 password may be a maximum of 30 characters.

```
cli> config network snmp v3 add username john password john1234 oid .1
permission ro
```

2. Activate and save your configuration.

Hosts

To configure hosts:

1. Add a host name with IP address.

   ```
   cli> config network hosttable add hostip <n.n.n.n> name [hostname]
   ```

   You may repeat this step as many times as necessary.

2. Activate and save your configuration.

TCP keepalive

The objective of this feature is to allow the console server to recognize when the socket client, SSH or Telnet goes down without closing the connection properly. The TCP engine of the console server sends a TCP keepalive message (ACK) to the client. If the maximum retry number is reached without an answer from the client, the connection is closed.

To configure TCP keepalive:

1. Configure the pool interval in milliseconds.

   ```
   cli> config physicalports all other tcpkeepalive <number>
   ```

2. Activate and save your configuration.

Firewall Configuration (IP Filtering)

IP filtering consists of blocking the passage of IP packets based on rules defined in the characteristics of the packets, such as the contents of the IP header, the input/output interface or the protocol. This feature is used mainly in firewall applications, which filter the packets that could crack the network system or generate unnecessary traffic.

Network Address Translation (NAT) allows the IP packets to be translated from local network to global network and vice-versa. This feature is particularly useful when there is demand for more IP addresses in the local network than available as global IP addresses. In the console server, this feature is used mainly for clustering (one master console server works as the interface between the global network and the slave console servers).

NOTE: The NAT table is not used with IPv6.

The console server uses the Linux utility iptables to set up, maintain and inspect both the filter and the NAT tables of IP packet rules in the Linux kernel. Besides filtering or translating packets, the iptables utility is able to count the packets which match a rule and to create logs for specific rules.

Structure of the iptables

The iptables are structured in three levels: table, chain and rule. A table may contain several chains and each chain may contain several rules.

Table

The table indicates how the iptables works. There are currently three independent tables supported by the iptables but only two are used.

- filter: This is the default table.
- nat: This table is consulted when a packet that creates a new connection is encountered.

Chain

Each table contains a number of built-in chains and may also contain user-defined chains. The built-in chains are called according to the type of packet. User-defined chains are called when a rule, matched by the packet, points to the chain. Each table has a specific set of built-in chains.

For the filter table:

- INPUT - For packets coming into the box itself.
- FORWARD - For packets being routed through the box.
- OUTPUT - For locally-generated packets.

For the nat table (IPv4 only):

- PREROUTING - For altering packets as soon as they come in.
- OUTPUT - For altering locally-generated packets as soon as they come in.
- POSTROUTING - For altering packets as they are about to go out.

Rule

Each chain has a sequence of rules. These rules contain:

- How the packet should appear in order to match the rule: Some information about the packet is checked according to the rule, such as the IP header, the input and output interfaces, the TCP flags and the protocol.
- What to do when the packet matches the rule: The packet may be accepted, blocked, logged or jumped to a user-defined chain. For the nat table, the packet may also have its source IP address and source port altered (for the POSTROUTING chain) or have the destination IP address and destination port altered (for the PREROUTING and OUTPUT chain).

When a chain is analyzed, the rules of this chain are reviewed one-by-one until the packet matches one rule. If no rule is found, the default action for that chain is taken.

Configuring IP tables

IPv4 Syntax

```
# iptables -command chain rule-specification [-t table] [options]
# iptables -E old-chain-name new-chain-name
```

where,

- table - May be filter or nat. If the option *-t* is not specified, the filter table is assumed.
- chain
  - For filter table: INPUT, OUTPUT, FORWARD or a user-created chain.
  - for nat table: PREROUTING, OUTPUT, POSTROUTING or a user-created chain.

IPv6 Syntax

```
# ip6tables -command chain rule-specification [-t table] [options]
# ip6tables -E old-chain-name new-chain-name
```

where,

- table - May only be a filter table. The option *-t* does not need to be specified.
- chain - INPUT, OUTPUT, FORWARD or a user-created chain.

NOTE: Fragmented packets cannot be filtered in IPv6 configurations.

Command

Only one command may be specified on the command line unless otherwise specified in Table 2.9.

Table 2.9: iptables Commands Options

Command	Description
-A --append	Append one or more rules to the end of the selected chain. When the source and/or destination names resolve to more than one address, a rule is added for each possible address combination.
-D --delete	Delete one or more rules from the selected chain. There are two versions of this command. The rule may be specified as a number in the chain (starting at 1 for the first rule) or as a rule to match.
-R --replace	Replace a rule in the selected chain. If the source and/or destination names resolve to multiple addresses, the command fails. Rules are numbered starting at 1.
-I --insert	Insert one or more rules in the selected chain as the given rule number. Thus if the rule number is 1, the rule or rules are inserted at the head of the chain. This is also the default if no rule number is specified.

Table 2.9: iptables Commands Options (Continued)

Command	Description
-L --list	List all rules in the selected chain. If no chain is selected, all chains are listed. It is legal to specify the -Z (zero) option as well, in which case the chain(s) are automatically listed and zeroed. The exact output is affected by the other arguments given.
-F --flush	Flush the selected chain. This is equivalent to deleting all the rules one-by-one.
-Z --zero	Zero the packet and byte counters in all chains. It is legal to specify the -L, --list (list) option as well, to see the counters immediately before they are cleared.
-N --new-chain	New chain. Create a new user-defined chain by the given name. There must be no target of that name already.
-X --delete-chain	Delete the specified user-defined chain. There must be no references to the chain. If there are, you must delete or replace the referring rules before the chain may be deleted. If no argument is given, it attempts to delete every non-built-in chain in the table.
-P --policy	Set the policy for the chain to the given target. Only non-user-defined chains may have policies and neither built-in nor user-defined chains may be policy targets.
-E --rename-chain	Rename the user-specified chain to the user-supplied name. This is cosmetic and has no effect on the structure of the table.
-h --help	Help. Gives a very brief description of the command syntax.

Rule specification

The following parameters define a rule specification as used in the add, delete, insert, replace and append commands.

Table 2.10: iptables Rules Specifications

Parameter	Description
-p	- -protocol[!]protocol The protocol of the rule or of the packet to check. The specified protocol may be one of TCP, UDP, ICMP (ICMPv6 for IPv6 configurations), ESP (IPv6 only), all, or it may be a numeric value, representing one of these protocols or a different one. A protocol name from /etc/protocols is also allowed. A ! argument before the protocol inverts the test. The number zero is equivalent to all. Protocol all matches with all protocols and is taken as default when this option is omitted.
-s	- -source[!]address[/mask] Source specification. Address may be either a hostname, a network name or a plain IP address. The mask may be either a network mask or a plain number, specifying the number of 1's at the left side of the network mask. Thus, a mask of 24 is equivalent to 255.255.255.0. A ! argument before the address specification inverts the sense of the address. The flag - -src is a convenient alias for this option.

Table 2.10: iptables Rules Specifications (Continued)

Parameter	Description
-d	- -destination[!]address[/mask] Destination specification. See the description of the -s (source) flag for a detailed description of the syntax. The flag - -dst is an alias for this option.
-j	- - jump target This specifies the target of the rule, for example, what to do if the packet matches it. The target may be a user-defined chain (other than the one this rule is in), one of the special built-in targets which decide the fate of the packet immediately, or an extension, see *Match extensions*. If this option is omitted in a rule, then matching the rule has no effect on the packet's fate, but the counters on the rule is incremental. The special built-in targets are: • ACCEPT means to let the packet through. • DROP means to drop the packet on the floor. • QUEUE means to pass the packet to userspace (if supported by the kernel). • RETURN means stop traversing this chain and resume at the next rule in the previous (calling) chain. If the end of a built-in chain is reached or a rule in a built-in chain with target RETURN is matched, the target specified by the chain policy determines the fate of the packet.
-i	- -in-interface[!][name] Optional name of an interface via which a packet is received (for packets entering the INPUT and FORWARD chains). When the ! argument is used before the interface name, the sense is inverted. If the interface name ends in a plus (+) then any interface which begins with this name matches. If this option is omitted, the string plus (+) is assumed, which matches with any interface name.
-o	- -out-interface[!][name] Optional name of an interface via which a packet is going to be sent (for packets entering the FORWARD and OUTPUT chains). When the ! argument is used before the interface name, the sense is inverted. If the interface name ends in a plus (+) then any interface which begins with this name matches. If this option is omitted, the string plus (+) is assumed, which matches with any interface name.
[!]	-f - -fragment This means that the rule only refers to second and further fragments of fragmented packets. Since there is no way to tell the source or destination ports of such a packet (or ICMP/ICMPv6 type), such a packet does not match any rules which specify them. When the ! argument precedes the -f flag, the rule only matches head fragments, or unfragmented packets.
-c	- -set-counters PKTS BYTES This enables the administrater to initialize the packet and byte counters of a rule (during INSERT, APPEND, REPLACE operations).
-v	- -verbose Verbose output. This option makes the list command show the interface address, the rule options, if any and the TOS masks. The packet and byte counters are also listed with the suffix K, M or G for 1000, 1,000,000 and 1,000,000,000 multipliers respectively (see the -x flag to change this). For appending, insertion, deletion and replacement, this causes detailed information on the rule or rules to be printed.

Table 2.10: iptables Rules Specifications (Continued)

Parameter	Description
-n	- -numeric Numeric output. IP addresses and port numbers are printed in numeric format. By default the program tries to display them as host names, network names or service, when applicable.
-x	- -exact Expand numbers. Display the exact value of the packet and byte counters, instead of only the rounded number in K's (multiples of 1000) M's (multiples of 1000K) or G's (multiples of 1000M). This option is specific to the -L command.
- -line-numbers	When listing rules, add line numbers to the beginning of each rule, corresponding to that rule's position in the chain.

Match extensions

Iptables may use extended packet matching modules. These are loaded in two ways: implicitly, when -p or - -protocol is specified, or with the -m or - -match option, followed by the matching module name; after these, various extra command line options become available, depending on the specific module.

TCP extensions

These extensions are loaded if the protocol specified is tcp or -m tcp is specified. It provides the following options.

Table 2.11: TCP Extensions

TCP extension	Description
--source-port [!] [port[:port]]	Source port or port range specification. This may either be a service name or a port number. Inclusive range may also be specified, using the format port:port. If the first port is omitted, 0 is assumed; if the last is omitted, “65535” is assumed. If the second port is greater then the first they is swapped. The flag - -sport is an alias for this option.
--destination-port [!] [port[:port]]	Destination port or port range specification. The flag - -dport is an alias for this option.
--tcp-flags [!] mask comp	Match when the TCP flags are as specified. The first argument is the flags which we should examine, written as a comma-separated list and the second argument is a comma-separated list of flags which must be set. Flags are: SYN ACK FIN RST URG PSH ALL NONE. Hence the command iptables -A FORWARD -p tcp - -tcp-flags SYN,ACK,FIN,RST SYN only matches packets with the SYN flag set and the ACK, FIN and RST flags unset.

Table 2.11: TCP Extensions (Continued)

TCP extension	Description
[!] --syn	Only match TCP packets with the SYN bit set and the ACK and FIN bits cleared. Such packets are used to request TCP connection initiation; for example, blocking such packets coming in an interface prevents incoming TCP connections, but outgoing TCP connections are unaffected. It is equivalent to - -tcp-flags SYN,RST,ACK SYN. If the ! flag precedes the - -syn, the sense of the option is inverted.
--tcp-option [!] number	Match if TCP option is set.

UDP extensions

These extensions are loaded if the protocol udp is specified or -m udp is specified. It provides the following options.

Table 2.12: UDP Extensions

UDP extension	Description
--source-port [!] [port[:port]]	Source port or port range specification. See the description of the - -source-port option of the TCP extension for details.
--destination-port [!] [port[:port]]	Destination port or port range specification. See the description of the - -destination-port option of the TCP extension for details.

ICMP extension

This extension is loaded if the protocol icmp is specified or -m icmp is specified. It provides the following option.

NOTE: For IPv6 configurations, the icmpv6 protocol is used.

Table 2.13: ICMP Extensions

ICMP extension	Description
--icmp-type [!] typename	This allows specification of the ICMP type, which may be a numeric ICMP type, or one of the ICMP type names shown by the command iptables p icmp -h

Multiport extension

This module matches a set of source or destination ports. Up to 15 ports may be specified. It may only be used in conjunction with -m tcp or -m udp.

Table 2.14: Multiport Extensions

Multiport extension	Description
--source-port [port[,port]]	Match if the source port is one of the given ports.
--destination-port [port[,port]]	Match if the destination port is one of the given ports.
--port [port[,port]]	Match if the both the source and destination port are equal to each other and to one of the given ports.

Target extensions

Iptables may use extended target modules. The following are included in the standard distribution.

LOG extensions

Turn on kernel logging of matching packets. When this option is set for a rule, the Linux kernel prints some information on all matching packets (like most IP header fields) via the kernel log.

Table 2.15: LOG Extensions

LOG extension	Description
--log-level level	Level of logging (numeric or see syslog.conf(5)).
--log-prefix prefix	Prefix log messages with the specified prefix; up to 29 letters long and useful for distinguishing messages in the logs.
--log-tcp-sequence	Log TCP sequence numbers. This is a security risk if the log is readable by users.
--log-tcp-options	Log options from the TCP packet header.
--log-ip-options	Log options from the IP packet header.

REJECT (filter table only)

This is used to send back an error packet in response to the matched packet, otherwise it is equivalent to DROP. This target is only valid in the INPUT, FORWARD and OUTPUT chains and

user-defined chains which are only called from those chains. Several options control the nature of the error packet returned.

Table 2.16: LOG Extension

LOG extension	Description
--reject-with type	The type given may be icmp-net-unreachable, icmp-host-unreachable, icmp-port-unreachable, icmp-proto-unreachable, icmp-net-prohibited or icmp-host-prohibited, which return the appropriate ICMP error message (port-unreachable is the default). The option echo-reply is also allowed; it may only be used for rules which specify an ICMP ping packet and generates a ping reply. Finally, the option tcp-reset may be used on rules which only match the TCP protocol. This causes a TCP RST packet to be sent back. This is mainly useful for blocking ident probes which frequently occur when sending mail to broken mail hosts (which won't accept your mail otherwise). **NOTE:** For IPv6 configurations, ICMPv6 types apply (such as icmpv6-net-unreachable).

SNAT (NAT table only, IPv4 only)

This target is only valid in the nat table, in the POSTROUTING chain. It specifies that the source address of the packet should be modified (and all future packets in this connection are also mangled) and rules should cease being examined. It takes one option.

Table 2.17: SNAT Target

SNAT target	Description
--to-source <ipaddr>[-<ipaddr>][:port-port]	This may specify a single new source IP address, an inclusive range of IP addresses and optionally, a port range (which is only valid if the rule also specifies -p tcp or -p udp). If no port range is specified, then source ports below 1024 are mapped to other ports below 1024. Those between 1024 and 1023 inclusive are mapped to ports below 1024 and other ports are mapped to 1024 or above. Where possible, no port alteration occurs.

DNAT (NAT table only, IPv4 only)

This target is only valid in the nat table, in the PREROUTING and OUTPUT chains and user-defined chains which are only called from those chains. It specifies that the destination address of

the packet should be modified (and all future packets in this connection are also mangled) and rules should cease being examined. It takes one option.

Table 2.18: DNAT Target

DNAT target	Description
--to-destination <ipaddr>[-<ipaddr>][:port-port]	This may specify a single new destination IP address, an inclusive range of IP addresses and optionally, a port range (which is only valid if the rule also specifies -p tcp or -p udp). If no port range is specified, then the destination port is never modified.

MASQUERADE (NAT table only, IPv4 only)

This target is only valid in the nat table, in the POSTROUTING chain. It should only be used with dynamically assigned IP (dialup) connections. If you have a static IP address, you should use the SNAT target. Masquerading is equivalent to specifying a mapping to the IP address of the interface the packet is going out on, but also has the effect that connections are forgotten when the interface goes down. This is the correct behavior when the next dialup is unlikely to have the same interface address (and hence any established connections are lost anyway). It supports one option.

Table 2.19: Masquerade Target

Target	Description
--to-ports <port>[-<port>]	Specifies a range of source ports to use. This parameter overrides the default SNAT source port-selection heuristics, see *SNAT (NAT table only, IPv4 only)*. This parameter is valid when the rule specifies -p tcp or -p udp.

REDIRECT (NAT table only, IPv4 only)

This target is only valid in the nat table, in the PREROUTING and OUTPUT chains and user-defined chains which are only called from those chains. It alters the destination IP address to send the packet to the machine itself (locally-generated packets are mapped to the 127.0.0.1 address). It supports one option.

Table 2.20: Redirect Target

Target	Description
--to-ports <port>[-<port>]	Specifies a range of source ports to use. This parameter overrides the default SNAT source port-selection heuristics, see *SNAT (NAT table only, IPv4 only)*. This parameter is valid when the rule specifies -p tcp or -p udp.

To configure firewall:

fwset script

Iptables rules are stored in /etc/network/firewall. The fwset script saves the iptables rules in /etc/network/firewall and saves it to Flash memory.

fwset restore

Restores the iptables' rules previously saved in /etc/network/firewall to their original configuration. This command is executed at boot to invoke the last saved configuration.

1. Execute fwset restore.
2. Add the required chains and rules. See *Configuring IP tables* on page 22.
3. Execute **iptables-save > /etc/network/firewall**.
4. Execute **fwset /etc/network/firewall to save the configuration in Flash memory.**

Static Routes

The Static Routes form allows you to manually add routes. The routing table defines which interface should transmit an IP packet based on destination IP information. Static routes are a quick and effective way to route data from one subnet to another. The static routing table may be viewed using either of the following commands.

```
[root@CAS root]# route
[root@CAS root]# netstat -rn
```

Routes may be added at the Linux shell prompt using the following command.

```
route [add|del] [-net|-host] target netmask nt_msk [gw gt_way] interf
```

Table 2.21: Routing Table Parameters

Parameter	Description
add|del	Routes may be either added or deleted. One of these options must be specified.
-net|-host	-net is for routes to a network and -host is for routes to a single host.
target IPv4: <nnn.nnn.nnn.nnn> IPv6: <IPv6_address>/<prefix_length>	Target is the IP address of the destination host or network.
netmask and *nt_msk*	Netmask and nt_mask are necessary only when subnetting is used. Otherwise, a mask appropriate to the target is assumed (IPv4 parameter only).
gw and *gt_way*	Specifies a gateway when applicable. The IP address or hostname of the gateway is specified by the gt_way parameter. **NOTE:** This can be an IPv4 or an IPv6 address.

Table 2.21: Routing Table Parameters (Continued)

Parameter	Description
interf	The interface to use for the route. Must be specified if a gateway is not. When a gateway is specified, the operating system determines which interface is to be used.

Use the following command to configure static routes. Refer to Table 2.22 for the list of parameters and the descriptions.

```
cli> config network stroutes add [parameter] <value>
```

Table 2.22: Static Routes Parameters and Values

Parameter	Value	Description
default	none	Used when there is no matching routing table.
gateway	IPv4: <nnn.nnn.nnn.nnn> or IPv6: <IPv6_address> (up to eight sets of four hexadecimal characters separated by colons (xxxx:xxxx. . .:xxxx:xxxx:)	Gateway IP address.
host	IPv4: <nnn.nnn.nnn.nnn> or IPv6: <IPv6_address> (up to eight sets of four hexadecimal characters separated by colons (xxxx:xxxx. . .:xxxx:xxxx:)	Route to a single host.
interface	<string>	Specify the network card that the packets come through.
metric	<number>	The number of routers that packets must pass through to reach the intended network.
netip	IPv4: <nnn.nnn.nnn.nnn> IPv6: <IPv6_address>/<prefix_length>	Route to a network.
netmask	<nnn.nnn.nnn.nnn>	Subnet mask (IPv4 parameter only).

Use the following command to delete a route.

```
cli> config network stroutes delete routenum <route number>
```

To configure static routes (example):

1. Add the default gateway 192.168.0.1.

   ```
   cli> config network stroutes add default gateway 192.168.0.1
   ```

2. Activate and save your configuration.

CHAPTER 3

Security

This chapter describes the procedures for configuring authentication service(s) that the console server and its connected devices use. Authentication is the process by which the system, or more specifically, an authentication service such as Kerberos, LDAP or TACACS+, verifies the identity of users as well as confirms receipt of communication to authorized recipients.

Security Profiles

The console server includes a set of security profiles that consist of predefined parameters to control access to the console server and its serial ports. To select a predefined or define a custom security profile refer to *Security Profiles* on page 15.

NOTE: As an additional security measure, all serial ports are disabled by default, which allows the administrator to enable and assign individual ports to users.

Authentication

The console server supports a number of authentication methods that may help the administrator with the user management. Authentication may be performed locally or with a remote server, such as RADIUS, TACACS+, LDAP, NIS or Kerberos. Should the negotiation process with the authentication server fail, an authentication security fallback mechanism is also employed. In such situations, the console server follows an alternate defined rule when the authentication server is down or does not authenticate the user.

CAUTION: If you set the authentication service in the console server to *NIS*, make sure that there is an entry for user id 0 (zero - the root user) in the NIS server. If you do not want an entry for user id 0 in the NIS server, set the authentication service in the console server to *Nis/Local*. Otherwise, root will not be able to ssh out of the console server, sudo will not work and the DSView® 3 software plug-in will not work.

NOTE: NIS does not work if Security Profile is set to Moderate or Secured. It only works if the Security Profile is Open.

To configure serial port authentication:

1. Execute the following command for one or multiple serial ports. Refer to Table 3.1 for authentication parameters.

```
cli> config physicalports <'all' or range/list[1-xx]> access authtype
[parameter]
```

2. Activate and save your configuration.

To configure general authentication to the console server:

1. Execute the following command to configure authentication. Refer to Table 3.1 for authentication parameters and fallback mechanisms.

   ```
   cli> config security authentication authtype [parameter]
   ```

2. Activate and save your configuration.

Table 3.1: Cyclades ACS 5000 Console Server Serial Port and General Authentication Methods

Authentication type	Parameter	Description
DSView	DSView DSView/Local DSViewDownLocal	Authentication is performed using DSView 3 management software. Local authentication is performed if the DSView 3 software fails or if the server is down.
Kerberos	Kerberos Kerberos/Local KerberosDownLocal	Authentication is performed using a Kerberos server. Local authentication is performed if Kerberos fails or if the Kerberos server is down.
LDAP	Ldap Ldap/Local LdapDownLocal	Authentication is performed using an LDAP server. Local authentication is performed if LDAP fails or if the LDAP server is down.
Local	Local Local/Nis Local/Radius Local/TacacsPlus	Authentication performed locally. NIS, Radius or TACACS+ authentication is used if the local authentication fails.
NIS	Nis Nis/Local NisDownLocal	NIS authentication is performed. Local authentication is performed if NIS fails or if the NIS authentication server is down.
OTP (Available for serial port authentication only.)	Otp Otp/Local	Uses the one time password (OTP) authentication method, or use local if OTP fails.
Radius	Radius Radius/Local RadiusDownLocal	Authentication is performed using a Radius server. Local authentication is performed if Radius fails or the Radius server is down.
TACACS+	TacacsPlus TacacsPlus/local TacacsPlusDownlocal	Authentication is performed using a TACACS+ authentication server. A local authentication is performed if TACACS+ fails or if the TACACS+ authentication server is down.

Table 3.1: Cyclades ACS 5000 Console Server Serial Port and General Authentication Methods

Authentication type	Parameter	Description
None (Available for serial port authentication only.)	none	Not a valid option when the serial port is configured for Power Management protocol. The system defaults to Local if no authentication type is selected.

To configure authentication servers:

1. Execute the following command to configure authentication server parameters. Refer to Table 3.2 for authentication servers parameters.

   ```
   cli> config security authentication [parameter] <value>
   ```

2. Activate and save your configuration.

NOTE: If IPv6 is enabled, then IP addresses in Table 3.2 can be entered in IPv6 format.

Table 3.2: Authentication Servers Parameters

Authentication Server	Parameter	Value
Kerberos	krbdomain krbserver	<domain name> <n.n.n.n>
LDAP	ldapbasedomain ldapserver	<ldapbasedomain> <n.n.n.n>
NIS	nisdomain nisserver	<domain name> <n.n.n.n>
Radius	radiusacctsvr1 radiusacctsvr2 radiusauthsvr1 radiusauthsvr2 radiusretries radiussecret radiussvctype radiustimeout	<n.n.n.n> <n.n.n.n> <n.n.n.n> <n.n.n.n> <number> <radiussecret> <yes|no> <number>
LDAP	secureldap	yes|no
TACACS+	tacplusacctsvr1 tacplusacctsvr2 tacplusauthsvr1 tacplusauthsvr2 tacplusraccess tacplusretries tacplussecret tacplustimeout	<n.n.n.n> <n.n.n.n> <n.n.n.n> <n.n.n.n> yes|no <number> <tacplussecret> <number>

User access to serial ports

To add groups and users:

1. Enter the following command to create user groups and add members, if required.

   `cli>` **`config security addgroup groupname <group name> usernames <[name_1, name_2, . . . name_n]>`**

2. Enter the following command to create users with administrative rights or limited access.

   `adduser` **`<user name> admin [yes|no] password <password> shell <shell> comments <comments>`**

Table 3.3: User Access Parameters

Parameter Level1	Parameter Level2	Value	Description
addgroup	groupname usernames	<group name> <list of user names separated by commas>	Add group and user members to manage access to connected servers.
delgroup	groupname	<groupname>	Delete group.
adduser	admin comments password shell username	yes|no <comments> <password> <shell> <user name>	Enable or disable administrative privileges. Specify user access to the Linux shell, CLI or none. Add user.
deluser	username	<user name>	Delete user.
loadkey	url username	<url> <username>	Using scp get the user's public key from the local database of the console server. <url> syntax: user@host:pathname
passwd	newpassword username	<password> <user name>	Change the user password.

To add groups and users (example):

- Add a group called FremontACS5000 that includes the users john and mary.

  `security>` **`addgroup groupname FremontACS5000 usernames john,mary`**

- Add a regular user (no admin privileges) named john with the password john1234.

  `security>` **`adduser username john admin no password john1234`**

- Load a key for the local “root” user accessed by root@192.168.0.1/home/key.

  ```
  security> loadkey username <username> url <url>
  security> loadkey username root url root@192.168.0.1/home/key
  ```

- Activate and save your configuration.

NIS Client

NIS (Network Information System) provides generic client-server database access facilities that can be used to distribute information. This makes the network appear as a single system, with the same accounts on all hosts. The objective of this feature is to allow the administrator to manage console server accounts on an NIS server.

The NIS client feature requires the files and commands listed in Table 3.4.

Table 3.4: NIS Client Requirements

File/Command	Description
/etc/yp.conf	This file contains the configuration used by ypbind.
/etc/domainname.conf	This file contains the NIS domain name (set by the command domainname).
/usr/sbin/ypbind	Finds the server for NIS domains and maintains the NIS binding information.
/usr/bin/ypwhich	Returns the name of the NIS server that supplies the NIS services.
/usr/bin/ypcat	Prints the values of all keys from the NIS database specified by map name.
/usr/bin/ypmatch	Prints the values of one or more keys from the NIS database specified by map name.
/usr/sbin/domainname	Shell script to read/write the NIS domain name.

NIS Client Configuration

1. Run the command domainname. Make sure that you have the NIS domain name set.

   ```
   # domainname [NIS domain name]
   ```

   Show or set the system's NIS/YP domain name, for example:

   ```
   # domainname avocent mycompany-nis
   ```

2. Edit the /etc/yp.conf file. Configure the NIS server. For example, if the NIS server has the IP address 192.168.160.110, add the following line to the file.

```
ypserver 192.168.160.110
```

3. Edit the /etc/nsswitch.conf file to include the NIS in the lookup order of the databases.
4. Configure the parameter <all/sxx>.authype as local.

To test the configuration:

1. Start with the following command.

   # **/usr/sbin/ypbind**

2. Display the NIS server name by running the following command.

   # **/usr/bin/ypwhich**

3. Display the all users entry by running the following command.

   # **/usr/bin/ypcat -t passwd.byname**

4. Display the user's entry in the NIS passwd file.

   # **/usr/bin/ypmatch -t <userid/username> passwd.byname**

If the preceding steps are performed successfully, change the /etc/inittab file by uncommenting the line that performs a ypbind upon startup.

nsswitch.conf

To use NIS to authenticate users, change the lines in /etc/nsswitch.conf that reference passwd, shadow and group.

The /etc/nsswitch.conf file has the following format.

```
<database> : <service> [<actions> <service>]
```

Table 3.5: nsswitch.conf Parameters

Parameter	Description
<database>	available: aliases, ethers, group, hosts, netgroup, network, passwd, protocols, publickey, rpc, services and shadow.
<service>	available: nis (use NIS version 2), dns (use Domain Name Service) and files (use the local files).
<actions>	Has this format: [<status> = <action>].
<status>	= SUCCESS, NOTFOUND, UNAVAIL or TRYAGAIN.
<action>	= return or continue.
SUCCESS	No error occurred and the desired entry is returned. The default action for this status is return.
NOTFOUND	The lookup process works fine, but the needed value was not found. The default action for this status is continue.

Table 3.5: nsswitch.conf Parameters (Continued)

Parameter	Description
UNAVAIL	The service is permanently unavailable.
TRYAGAIN	The service is temporarily unavailable.

The following examples illustrate the use of NIS to authenticate users.

- Authenticate the user in the local database; if the user is not found, then use NIS.

```
passwd: files nis
shadow: files nis
group: files nis
```

- Authenticate the user using NIS; if the user is not found, then use the local database.

```
passwd: nis files
shadow: nis files
group: nis files
```

- Authenticate the user using NIS; if the user is not found or the NIS server is down, use the local database.

```
passwd: nis [UNAVAIL=continue TRYAGAIN=continue] files
shadow: nis [UNAVAIL=continue TRYAGAIN=continue] files
group: nis [UNAVAIL=continue TRYAGAIN=continue] files
```

Kerberos Authentication

Kerberos is a network authentication protocol designed for use on unsecured networks, based on the key distribution model. It allows individuals communicating over a network to prove their identity to each other while preventing eavesdropping or replay attacks. It provides detection of modification and prevention of unauthorized reading.

Kerberos server authentication with tickets support

The console server has support to interact on a kerberized network. On a kerberized network, the Kerberos database contains principals and keys (for users, keys are derived from passwords). The Kerberos database also contains keys for all of the network services.

When a user on a kerberized network logs in to the workstation, the principal is sent to the Key Distribution Center (KDC) as a request for a Ticket Granting Ticket (TGT). This request may be sent by the login program so that it is transparent to the user, or may be sent by the kinit program after the user logs in.

The KDC checks for the principal in its database. If the principal is found, the KDC creates a TGT, encrypts it using the user's key and sends it back to the user.

The login program or kinit decrypts the TGT using the user's key, which is computed from the user's password. The TGT, which is set to expire after a certain period of time, is stored in the credentials cache. An expiration time is set so that a compromised TGT may only be used for a certain period of time, usually eight hours, unlike a compromised password, which could be used until changed. The user does not have to re-enter the password until the TGT expires or a new session is started.

When the user needs access to a network service, the client uses the TGT to request a ticket for the service from the Ticket Granting Service (TGS), which runs on the KDC. The TGS issues a ticket for the desired service, which is used to authenticate the user.

Configuring the console server to use Kerberos tickets authentication

The following procedure describes the console server's configuration, assuming that the kerberos server with ticket support is properly configured with the following parameters.

- Principal: john
- Host: acs48.cyclades.com

To configure the console server for SSH:

1. Configure and start an NTP server. Configuration must be synchronized with an NTP server. To configure an NTP server, see *To configure an NTP server:* on page 85.
2. Configure authentication type and protocol in the /etc/portslave/pslave.conf file with the following parameters.

   ```
   all.authtype local
   all.protocol socket_ssh.
   ```

3. Activate and save the configuration.

   # **runconf**

   # **saveconf**

4. Add a user with the same name as the principal in the Kerberos server.

   # **adduser john**

5. Configure the krb5.conf file. The /etc/krb5.conf file must be exactly the same as the one that is in the Kerberos server. It is highly recommended to copy it directly from the server, instead of editing it. To copy using scp, execute the following command.

   # **scp root@kerberos-server.cyclades.com:/etc/krb5.conf /etc/krb5.conf**

6. Extract the host that is in the Kerberos server database to the console server.

   # **kadmin -p admin/admin**

   Where the first admin is the service and the second admin is the user.

   This prompts a Kerberos server menu. To extract the configured hosts run the following commands in the kadmin menu.

```
kadmin: ktadd host/acs48-2.cyclades.com
kadmin: q
```

To list all configured hosts in the Kerberos server, run the following command, which displays all hosts added through the ktadd command in the Kerberos server.

```
# klist -k
```

7. Configure hostname and domain name.

```
# hostname acs48-2
# domainname cyclades.com
```

To access the console server through rlogin and Telnet:

In addition to performing the steps described in *To configure the console server for SSH:* on page 40, make the following configuration changes.

1. Configure the /etc/inetd.conf file by uncommenting the following line lines.

```
#KERBEROS SERVICES
klogin stream tcp nowait root /usr/sbin/tcpd /usr/local/sbin/klogind
-ki
telnet stream tcp nowait root /usr/sbin/tcpd /usr/local/sbin/telnetd
```

2. Restart the inetd service.

```
# daemon.sh restart NET
```

3. Save the configuration.

```
# saveconf
```

To test the configuration:

1. The client must have a kerberized SSH. In addition, configure the following parameters in the etc/ssh/ssh_config file.

```
GSSAPIAuthentication yes
GSSAPICleanupCreds yes
```

2. The client must have the same krb5.conf file in the Kerberos server.

```
# scp root@kerberos-server.cyclades.com:/etc/krb5.conf /etc/krb5.conf
```

3. Request the ticket from the Kerberos server.

```
# kinit -f -p john
Password for john@CYCLADES.COM: ******
```

You are prompted to insert the principal password, which is in the Kerberos server database.

4. Check to see if the ticket was received successfully.

```
# klist
```

5. Connect from the client to the console server through SSH.

```
# ssh john@acs5048-2.cyclades.com
```

6. Open an SSH session to one of the console server's ports.

```
# ssh john:7001@acs5048-2.cyclades.com
```

7. RLOGIN to the console server with forwardable tickets.

```
# rlogin -l john acs5048-2.cyclades.com -F
```

8. Telnet to the console serverwith forwardable tickets.

```
# telnet -l john acs5048-2.cyclades.com -F
```

Kerberos server authentication

1. Open the /etc/portslave/pslave.conf file.

```
# vi /etc/portslave/pslave.conf
```

2. Change the values of the following parameters.

```
all.authtype     kerberos
all.protocol     socket_ssh ##or socket_server or socket_server_ssh
```

To use the Telnet protocol to access the serial ports, set the all.protocol parameter to socket_server.

To use both Telnet and SSH to access the unit, set the all.protocol parameter to socket_server_ssh.

3. Edit the /etc/krb5.conf file.

```
# vi /etc/krb5.conf
```

All changes required in this file are related to the network domain. Substitute all listed parameters that are configured with cyclades.com with the corresponding domain of your network.

4. Activate your changes.

```
# runconf
```

5. Test the configuration.
   a. Access a serial port using the Telnet protocol, for example:

```
# telnet 192.168.0.1 7001
```

   b. Log in with the user and password previously configured in the Kerberos server.
   c. In the console server, run the following command.

```
# w
```

6. Save your changes.

   # **saveconf**

LDAP Authentication

To configue LDAP authentication on the console server:

1. Execute the following command. Refer to Table 3.6 for authentication parameters.

   cli> **config security authentication [parameter] <value>**

2. Activate and save your configuration.

Table 3.6: LDAP Authentication Parameters

Parameter	Value	Description
ldapbasedomain	<ldapbasedomain>	Distinguished name of the search base. dc=cyclades,dc=com
ldapserver	<n.n.n.n>	LDAP server IP address or name.
secureldap	yes|no	To use secure LDAP.

Group Authorization

This feature enables the group information retrieval from the authentication servers TACACS+, RADIUS and LDAP. It adds another layer of security by adding a network-based authorization. It retrieves the group information from the authentication server and performs an authorization through the console server.

TACACS+ authorization on serial ports

By enabling the raccess parameter, administrators implement an additional level of security checking. After each user is successfully authenticated through the standard login procedure, the console server uses TACACS+ to authorize user access to specific serial ports.

By default, the raccess parameter is disabled. When enabled, users are denied access unless they have the proper authorization, which must be set on the TACACS+ server itself.

To configure TACACS+ authorization on serial ports:

1. Enable raccess authorization parameter.

   cli> **config security authentication tacplusraccess [yes|no]**

2. Configure serial ports for user or group access.

   cli> **config physicalports <'all' or range/list[1-xx]> access users/ groups <list of users or group names separated by commas>**

3. Activate and save your configuration.

To configure a TACACS+ authentication server:

1. On the server, add raccess service to the user configuration.
2. Define to which group or groups the user belongs.

```
user = <username>{
global = cleartext “<password>”
service = raccess{
group_name = <Group1>[,<Group2,...,GroupN>];
}
}
```

To configure user permission on the TACACS+ server:

1. On the TACACS+ server, open the file /etc/tacacs/tac_plus.cfg.

NOTE: The location of this configuration file may be different on your Linux distribution.

2. Edit the parameters as per the following example. Refer to Table 3.7 for descriptions.

```
user = tomj{
    name = "Tom Jones"
    service = raccess {
       port1 = LAB2/ttyS2
       port2 = 192.168.0.1/ttyS1
       port3 = CAS/ttyS1
       port4 = 172.32.20.10/ttyS6
       port5 = LAB1/ttyS7
       port6 = Knuth/ttyS16
    }
}
```

Table 3.7: Parameters for Specifying User Authorization on a TACACS+ Server

Parameter	Description
user = <*username*>	Defines the username as specified on the console server.
name = <*"optional description"*>	To specify additional information about the user (optional). This parameter must include quotes. The maximum number of characters allowed is 256. Adding more than 256 characters stops the server from restarting and produces a FAILED message at the time of authorization.

Table 3.7: Parameters for Specifying User Authorization on a TACACS+ Server (Continued)

Parameter	Description
service = <*authorization method*>	Specifies the authorization method used and whether the user is allowed or denied access when the raccess parameter is set on the console server. Only users who have this parameter set to raccess have authorization to access the specified ports.
port<#> = <*ACS5000*>/ <*Port*>	Specify which serial ports on the console server the user has authorization to access. port# is a sequential label used by the console server. <ACS5000> is the name or IP address of the console server box. <Port> is the serial port the user may access on the specified console server.

3. On the console server, use the CLI utility to edit the parameters described in the following table.

   `cli>` **`config security authentication [parameter] <value>`**

Table 3.8: TACACS+ Configuration Parameters

Parameter	Value	Description
tacplusauthsvr1	<n.n.n.n>	This address indicates the location of the TACACS+ authentication server. A second TACACS+ authentication server may be configured with the parameter tacplusauthsvr2.
tacplusacctsvr1	<n.n.n.n>	This address indicates the location of the TACACS+ accounting server, which may be used to track how long users are connected after being authorized by the authentication server. Its use is optional. If this parameter is not used, accounting is not performed. If the same server is used for authentication and accounting, both parameters must be filled with the same address. A second TACACS+ accounting server may be configured with the parameter accthost2.
tacplussecret	<tacplussecret>	This is the shared secret (password) necessary for communication between the console server and the TACACS+ servers.
tacplusraccess	yes|no	This is raccess authorization on the TACACS+ server. Should be enabled for authorization on serial ports.
tacplustimeout	<number>	This is the time-out (in seconds) for a TACACS+ authentication query to be answered.
tacplusretries	<number>	Defines the number of times each TACACS+ server is tried before another is contacted. The first server *authhost1* is tried for the specified number of times, before the second *authhost2*, if configured, is contacted and tried for the specified number of times. If the second server fails to respond, TACACS+ authentication fails.

To configure a RADIUS authentication server:

1. On the Radius server, edit /etc/raddb/users and add a new string attribute (ATTRIBUTE Framed-Filter-Id) similar to the following example.

```
groupuser1 Auth-Type= Local, Password =”xxxx”
Service-Type=Callback-Framed-User,
Callback-Number=”305”,
Framed-Protocol=PPP,
Framed-Filter-Id
Framed-Filter-Id=”group_name=<Group1>[,<Group2>,...,<GroupN>]”;
Fall-Through=No
```

If the Frame-Filter-Id already exists, add the group_name to the string starting with a colon (:).

2. On the console server, use the CLI utility to edit the parameters described in the following table.

```
cli> config security authentication [parameter] <value>
```

Table 3.9: Radius Configuration Parameters

Parameter	Value	Description
radiusauthsvr1	<n.n.n.n>	This address indicates the location of the Radius authentication server. A second Radius authentication server may be configured with the parameter radiusauthsvr2.
radiusacctsvr1	<n.n.n.n>	This address indicates the location of the Radius accounting server, which may be used to track how long users are connected after being authorized by the authentication server. Its use is optional. If this parameter is not used, accounting cannot be performed. If the same server is used for authentication and accounting, both parameters must be filled with the same address. A second Radius accounting server may be configured with the parameter *accthost2*.
radiussecret	<radiussecret>	This is the shared secret (password) necessary for communication between the console server and the Radius servers.
radiustimeout	<number>	This is the timeout (in seconds) for a Radius authentication query to be answered.
radiusretries	<number>	Defines the number of times each Radius server is tried before another is contacted. The first server radiusauthhost1 is tried for the specified number of times, before the second radiusauthhost*2*, if configured, is contacted and tried for the specified number of times. If the second server fails to respond, Radius authentication fails.

Table 3.9: Radius Configuration Parameters (Continued)

Parameter	Value	Description
radiussvctype	<yes/no>	Set to “no” to authorize the ACS console server to retrieve the level of user based on the group_name attribute sent by the RADIUS server. Set to “yes” to authorize the ACS console server to retrieve the level of the user (admin or regular) based on the Service-Type attribute from the RADIUS server.

To configure RADIUS authorization on the console server to access the serial ports:

1. In CLI mode, enter the following string.

   ```
   cli> config physicalports <serial port number> access users/groups
   <list of users or group names separated by commas>
   ```

2. Activate and save your configuration.

To configure an LDAP authentication server:

On the LDAP server, edit the info attribute for the user and add the following syntax.

```
info: group_name=<Group1>[,<Group2>,...,<GroupN>];
```

To configure LDAP authorization on the console server to access the serial ports:

1. In CLI mode, enter the following string.

   ```
   cli> config physicalports <'all' or range/list[1-xx]> access users/
   groups <list of users or group names separated by commas>
   ```

2. Activate and save your configuration.

One Time Password (OTP) Authentication

This section describes the procedures required to set up and configure OTP (one-time password) for OTP authentication type. OPIE (one-time passwords in everything) software on the Cyclades ACS 5000 console server supports OTP authentication to access the serial ports.

OPIE software on the console server supports the OTP authentication method and the OTP/Local fallback option for serial ports. The OTP authentication method is supported for dial-ins through external modem.

See http://www.freebsd.org/doc/en/books/handbook/one-time-passwords.html for more details about OTP.

OTP authentication configuration tasks

Console server administrators must perform the following tasks to set up and configure OTP.

- Mount the OTP database on any of the following storage units.

  - The main memory on the console server
  - NFS-mounted directory
- Configure OTP for each user. The console server administrator must make sure each user who needs to use OTP has a local account on the console server and is registered with the OTP system.
- Configure a serial port configured as PPP and with an external modem for OTP authentication. You may use WMI or the CLI utility to configure a serial port for OTP.

To set up and configure an OTP database:

1. Open a console window and log in to the console server as root.
2. Execute the following command to configure the OTP database.

   # **do_create_otpdb**

3. Enter the desired location where you want the OTP database stored. The following table shows the available options.

Table 3.10: OTP Database Location Options

Location	Notes
Local	Locally on the console server memory. The file is available during run-time. It will not be saved in the FLASH automatically. (The file /etc/opiekeys needs to be added in the /etc/config_files file to be saved in the FLASH by saveconf command.)
NFS	host:path host - DNS name or IP address of the NFS server. path - Directory shared by the NFS server.

4. Enable OTP. By default OTP is disabled.
5. The OTP database is mounted once you enable OTP.

Proceed to the following section to register users and generate OTP passwords.

To register users for OTP:

The following procedures should be performed for each user who requires OTP authentication. The following example demonstrates how to add and register a new user to the console server.

1. Log in locally through the console server port as root or use ssh to log in remotely.
2. Execute the adduser command. If a user account exists in the console server, skip this step and proceed to step 3 to register the user for OTP.

   # **adduser [username]**

   New password: users_passwd

```
Re-enter new password: users_passwd
```

3. Execute the opiepasswd command to register a user and generate a default OPIE key. This command initializes the system information to allow using OPIE login.

NOTE: You may use the -c option (console mode) if you have secure access to the console server. Running OPIE commands through an unsecured connection may reveal your password and compromise security.

Using opiepasswd from the console

The following information displays when you execute the opiepasswd command from the console with a -c option. The system prompts you to enter a new secret pass phrase and proceeds to generate default OPIE sequence number 499 and a key from the first two letters of the hostname (kv), a pseudo random number (6178) and a password comprised of six words. In the following example, 499 KV6178 is the OPIE key and the password is COMB YANK BARD SLOT AS USER.

```
opiepasswd -c peter
Adding peter:
Only use this method from the console; NEVER from remote. If you are
using telnet, xterm, or a dial-in, type ^C now or exit with no
password.
Then run opiepasswd without the -c parameter.
Using MD5 to compute responses.

Enter new secret pass phrase: peters passphrase
Again new secret pass phrase: peters passphrase

ID peter OTP key is 499 KV6178
COMB YANK BARD SLOT AS USER
```

Using opiepasswd from remote

When you execute the opiepasswd command securely from a remote system, you need an OTP generator (calculator) to obtain the OTP password. This initial sequence and its password are used to generate the hash number stored in the OTP database. Contact your system administrator to obtain an OTP calculator.

```
# opiepasswd john
Adding john:
You need the response from an OTP generator.
New secret pass phrase:
        otp-md5 499 KV3881
        Response:JOE FEE JUTE HARK BANE FAR
ID  OTP key is 499 KV3881
JOE FEE JUTE HARK BANE FAR
```

To generate OTP passwords:

1. Execute the command opiekey to generate passwords for the users.

NOTE: Do not execute the opiekey command through dial-in or an unsecured remote connection such as Telnet.

The following example uses MD5 (-5 option) to verify data integrity. The -n <count> option followed by the sequence number 498 generates 5 passwords ending with number 498.

```
# opiekey -5 -n 5 498 KV6178

Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase: john's secret pass phrase
494: HOST DRUG CLAN NARY HILT BULB
495: DUG JET CAIN SKIN SIGN BRAE
496: ALOE DUEL HUB SIT AMMO MIN
497: REEK KEN RECK CUT NEWS AMY
498: ALGA DEAD PUN FLUB LYRA LEN
```

2. Give the OTP username, secret pass phrase and the OTP passwords generated in this procedure to the user.

Shadow Passwords

The console server has support for shadow passwords, which enhances the security of the system authentication files. Shadow Passwords are enabled by default.

Digital Certificates

Certificate for HTTP security

The following procedure enables you to obtain a Signed Digital Certificate. A certificate for the HTTP security is created by a Certification Authority (CA). Certificates are most commonly obtained through generating public and private keys using a public key algorithm like RSA or X.509. The keys may be generated by using a key generator software.

To obtain a signed digital certificate:

1. Enter the OpenSSL command. Key generation may be done using the OpenSSL package using the following command:

```
# openssl req -new -nodes -keyout private.key -out public.csr
```

The Certificate Signing Request (CSR) generated by the command contains some personal or corporate information and its public key.

Table 3.11: Required Information for the OpenSSL Package (etc/openssl.conf file by default)

Parameter	Description
Country Name (2 letter code)	The country code consisting of two letters.
State or Province Name (full name)	Provide the full name (not the code) of the state.
Locality Name	Enter the name of your city.
Organization Name	Organization for which you are obtaining the certificate.
Organizational Unit Name	Department or section where you work.
Common Name	Name of the server where the certificate should be installed.
Email Address	Your email address or the administrator's email address.

2. Submit the CSR to CA for approval. This service may be requested by accessing the CA's web site. Visit pki-page.org for a list of CAs.
3. Once approved, CA sends the certificate file to the originator. The certificate is stored on a directory server. The following procedures describe the certificate installation process.

To install the certificate on the web server:

1. Log in to the console server.
2. Create the /etc/CA/server.pem file by combining the certificate with the private key.

   # **cat Cert.cer private.key > /etc/CA/server.pem**
3. Copy the certificate to the /etc/CA/cert.pem file.

   # **cp Cert.cer /etc/CA/cert.pem**
4. Save the configuration in Flash.

   # **saveconf**
5. Reboot the console server for the certificate to take effect.

User configured digital certificate

The console server generates its own self-signed SSL certificate for HTTPs using OpenSSL. It is highly recommended that you use the "openssl" tool to generate a self-signed certificate and replace the console server's generated certificate.

To generate a self-signed certificate:

1. Open the /etc/req_key file and update the user data with your organization specific data.

```
# vi /etc/req_key
[ req ]
default_bits            = 1024
distinguished_name      = cyclades
prompt                  = no
x509_extensions         = x509v3

[ cyclades ]
C               = US
ST              = CA
L               = Fremont
O               = Cyclades Corporation
OU              = R&D
CN              = www.cyclades.com
emailAddress    = support@cyclades.com

[ x509v3 ]
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid:always,issuer:always
basicConstraints        = CA:true
nsComment               = "This is just a TEST certificate.
nsCertType              = server, sslCA
```

2. Remove the files /etc/ca/*.pem
3. Execute the following script.

   # **/bin/firstkssl.sh**

4. Reboot the console server or restart web manager.

X.509 certificate on SSH

The OpenSSH software included with the console server has support for X.509 certificates. The administrator must activate and configure SSH to use X.509.

To configure an X.509 certificate for SSH:

1. Enter the following command to configure an X.509 certificate. See Table 3.12 for the list of parameters.

   ```
   cli> config security profile custom ssh ssh_x509 [parameter] <value>
   ```

Table 3.12: X.509 Certificate Parameters

Parameter	Value
CA_file	<path and filename of CA certificate>
hostkey	<path and filename of hostkeys>
authorizedkeys	path and filename of authorized keys>

2. Activate and save your configuration.

The following is an example on how to configure an X.509 certificate.

```
ssh_x509> cp CA_file /etc/ssh/ca-bundle.crt
ssh_x509> cp hostkey /etc/ssh/hostkey
ssh_x509> cp authorizedkeys /etc/ssh/authorized_keys
ssh_x509> chmod 600 /etc/ssh/authorized_keys
ssh_x509> chmod 755 /
cli> config runconfig
cli> config savetoflash
```

NOTE: An X.509 certificate for SSH may also be configured by executing the following script at the command prompt, **# ssh_act_x509**.

To connect to the console server and serial ports using an SSH X.509 certificate:

1. Configure an X.509 certificate for SSH.
2. Configure the client you need to access with an X.509 certificate.
3. Copy the certificate files to the console server.

   To verify that the file was copied, run the following command at the prompt.

   ```
   [root@acs48 root]# ls -l /etc/ssh/ca/ca-bundle.crt
   [root@acs48 root]# ls -l /etc/ssh/hostkey
   ```

4. Configure the serial ports for socket_ssh protocol and assign the IP address of the connected device.

CHAPTER 4

Accessing Connected Devices

This chapter describes set up and configuration parameters for accessing serial ports and connected devices.

Connection Profiles and Protocols

The following table describes each connection profile and supported protocols.

Table 4.1: Connection Profiles and Protocols

Connection Profile	Supported Protocol	Description
Console Access Server (CAS)	Telnet SSH TelnetSSH Raw	Configure when a serial port is connected to the console port of a server.
Terminal Server (TS)	Telnet SSHv1 SSHv2 Local Terminal Raw Socket	Configure when a terminal is connected to the console port of a server.
Bidirectional Telnet	Telnet (CAS) Telnet (TS)	Supports both CAS profile Telnet connection and TS profile menu shell. Both connection protocols are supported on one port; however, connections can not be opened simultaneously.
Modem (RAS)	PPP PPP-No Auth SLIP CSLIP	Configure when a modem is connected to a serial port.
Power Management	Power Management	Configure when a power management device is connected to a serial port.

Serial ports general parameters

To configure general parameters:

1. Execute the following command for one or multiple serial ports. Refer to Table 4.2 for port configuration parameters.

   ```
   cli> config physicalports <'all' or range/list[1-4] general
   [parameter] <value>
   ```

2. Activate and save your configuration.

Table 4.2: Serial Port General Configuration Parameters

Parameter	Value	Description
alias	<server alias>	To name a server connected to the serial port.
datasize	<number[5-8]>	To configure number of bits per character.
dcdstate	disregard|regard	To enable or disable Data Carrier Detect (DCD).
flow	hard|none|soft	To set the flow control.
parity	even|none|odd	To configure parity.
pmsessions	none ssh ssh_telent telnet	To select a connection method to PM IPDU through the serial port, in order to execute PM commands.
protocol	bidirectionaltelnet consoleraw consolessh consoletelnet consoletelnetssh cslip local pm ppp pppnoauth rawsocket slip sshv1 sshv2 telnet	To configure the serial ports connection protocol. See *Connection Profiles and Protocols* on page 55 for a description of each connection profile.
speed	<baud rate>	To configure the serial port speed.
stopbits	<number[1-2]>	To configure the number of stop bits.

To configure other configuration parameters:

1. Execute the following command for one or multiple serial ports. Refer to Table 4.3 for configuration parameters.

   `cli>` **`config physicalports <'all' or range/list[1-4] other [parameter] <value>`**

2. Activate and save your configuration.

Table 4.3: Other Serial Port Configuration Parameters

Parameter	Value	Description
banner	<login banner>	To set the banner that is displayed when you connect to a serial port. Text should be entered in double quotes (“ ”).
breakinterval	<number>	To set break interval in milliseconds (ms). Usually 250 to 500 milliseconds.
breaksequence	<break sequence>	To set the break sequence. Usually a character sequence, ~break (**Ctrl+b**).
host	<hostname>	IP address or the name of the server to which you are connecting.
idletimeout	<number>	To configure idle time-out, which is the maximum time (in seconds) that a session may be idle before the user is logged off.
portip	<n.n.n.n>	To configure an IP alias to the serial port.
sttyoptions	<stty options>	To set terminal options.
tcpkeepalive	<number>	To configure poll interval in milliseconds (ms). Specifies the time interval between the periodic polling to check client processes and connectivity.
tcpport	<number>	To configure socket port number. Four-digit values are valid for this parameter, for example 7001.
terminaltype	<terminal type>	To configure the terminal type when connecting to a host system.
winems	yes|no	Enables or disable Windows Emergency Management Services (EMS).
sconfkey	<serial configuration mode key>	The key sequence to perform serial port line configuration by a menu driven in the session opened against the serial port. Users that have access rights to the serial port will be authorized to perform this operation.

To open and close a Telnet session to a serial port:

```
# telnet [hostname] [TCP port number]
```

Table 4.4: Telnet Session Configuration Parameters

Parameter	Description
hostname	Workstation name or its IP address.
TCP port number	TCP port number assigned to the serial port.

To close a Telnet session, press the hotkey defined for the Telnet client. The default is **Ctrl+]**.

To open and close an SSH session to a serial port:

```
# ssh -l [username]:[server] [hostname]
```

Table 4.5: SSH Session Configuration Parameters

Parameter	Description
username	User configured to access the serial port. It is present either in the local database or in an authentication server such as Radius or LDAP.
server	TCP port number assigned to a serial port (for example 7001), pool of ports (for example 3000), the alias for the server connected to that serial port or the alias of a pool of ports.
hostname	Workstation name or its IP address.

To close an SSH session, press the hotkey defined for the SSH client followed by a dot (.). The default is tilde (**~**).

NOTE: Enter the escape character followed by a dot (**.**) at the beginning of a line to close the SSH session.

Accessing serial ports using ts_menu

The ts_menu is an application to facilitate connection to the serial ports. The following are the methods of executing the ts_menu command.

- Calling ts_menu without specifying arguments.
- Calling ts_menu with command line arguments.
- Using CLI to call ts_menu.

Calling ts_menu without specific parameters

To access the serial port configured for Telnet or SSH, enter ts_menu at the shell prompt. The server's aliases or serial ports are displayed as options to start a connection.

Calling ts_menu with specific parameters

```
# ts_menu -u <user> [-l[c]] [-ro] [-s] [-auth] <console port>
```

Table 4.6: ts_menu Parameters

Parameter	Description
`-u <user>`	Invokes ts_menu as the user named by <user>. This requires a password to be entered. The user has access only to the authorized serial ports.
`-l[c]`	Generates a list of ports to which a user has access. Port aliases are shown if defined. For remote ports (clusters), if port alias is not defined they are shown as ip_addr:port (ip_addr referring to the slave console server). The default displays ports in alphabetical order, but if c flag is specified, the listing is sorted by the console server (master unit showing first).
`-ro`	Invokes ts_menu in read-only mode. You may connect to any port to which you have access in read-only mode.
`-s`	Invokes ts_menu in a way that all ports (including slave) are presented in a single list sorted in alphabetical order.
`-auth`	For backward compatibility. This option makes the new ts_menu implementation behave as the old one so that authentication is performed again to access each port.
`<console port>`	If issued, produces a direct connection to that port. If you have no access rights to the port or if the port does not exist, the application returns a console not found message and terminates. The console port may be the port alias or the port number. If you are trying to access a clustered port, the port number must include a reference to the slave console server as host:port. Host is the slave hostname or IP address.
`-p`	Display TCP port.
`-P`	Use TCP port instead of IP address.
`-i`	Display Local IP assigned to the serial port.
`-s`	Show the ports in a sorted order.
`-u <name>`	Username to be used in SSH/Telnet command.
`-U`	Always ask for a username.
`-e <[^]char>`	Escape character used by Telnet or SSH.

To close the session from ts_menu (local):

1. Enter the escape character shown when you connect to the port. In character/text mode, the Escape character is **Ctrl+]**.
2. Console escape commands are displayed.

Table 4.7: Console Escape Commands

Command	Description
l	go to line mode
c	go to character mode
z	suspend telnet
b	send break
t	toggle binary
e	exit telnet

3. Press -e to exit from the session and return to the original menu.
4. Select the exit option to return to the shell prompt.

To close the session from ts_menu (Telnet/SSH):

Unless a different escape character is used for closing your Telnet/SSH session, you may close your entire Telnet session. To specify a different character, connect to your unit and use the -e option. For example, to set **Ctrl+?** as the escape character, type:

```
# telnet -e ^? 192.168.160.10
# ssh -e ^? user1@192.168.160.10
```

To exit from an entire Telnet session, type the escape character. For a SSH session, type the escape character plus dot (.).

NOTE: To close an SSH session the escape character followed by a dot (.) must be entered at the beginning of a line.

To call ts_menu from CLI:

1. Execute the following command from the CLI prompt. Refer to Table 4.8 for configuration parameters.

   ```
   cli> applications connect [parameter] <value>
   ```

2. Activate and save your configuration.

Table 4.8: ts_menu Configuration Parameters

Parameter	Value	Description
consolename	<consolename>	Name of the serial port to which you need to connect.
list		To display a list of the available serial ports.
readonly		To connect to the console of a server in read-only mode. Add the serial port name parameter: cli> applications connect readonly consolename <consolename>

Configuration examples

Console Access Server (CAS) profile

With the console server set up as a CAS profile, you may access a connected server's serial console port from a workstation on the network. There is no authentication by default, but the system may be configured for an authentication server such as Radius, LDAP or a local database.

Figure 4.1 displays an example of a CAS environment, and descriptions follow in Table 4.9. This configuration example has local authentication and serially connected workstations.

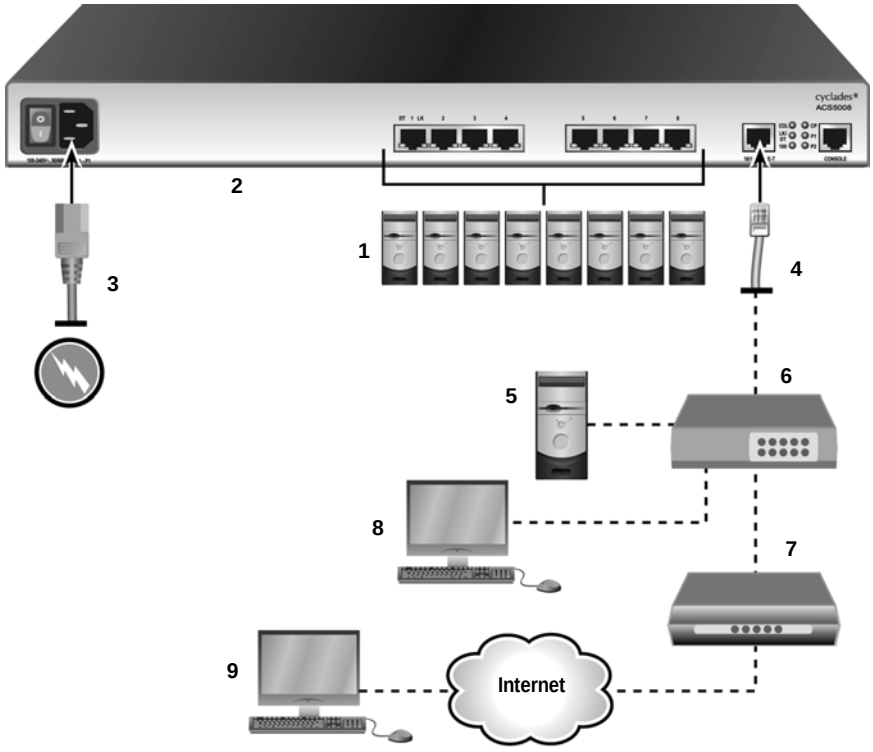

Figure 4.1: Example of CAS Configuration with Local Authentication

Table 4.9: Example of CAS Configuration with Local Authentication Descriptions

Item	Description	Item	Description
1	Servers on Serial Ports	**6**	Ethernet Hub or Switch
2	Cyclades ACS 5000 Console Server	**7**	Ethernet Router
3	Power Cable	**8**	Local User
4	Ethernet CAT 5 Cable	**9**	User
5	Local Network Server		

Figure 4.2 displays another configuration example for remote and local authentication, data buffering and remote access. Descriptions follow in Table 4.10.

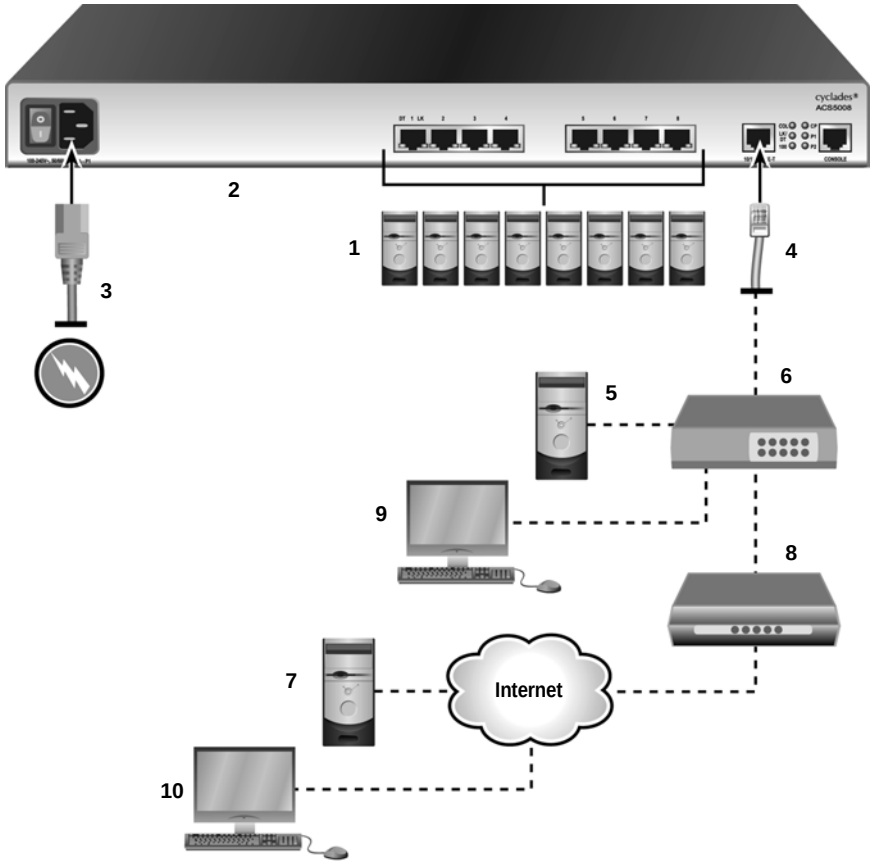

Figure 4.2: Example of CAS Configuration with Local and Remote Authentication

Table 4.10: Example of CAS Configuration with Local and Remote Authentication Descriptions

Item	Description	Item	Description
1	Servers on Serial Ports	**6**	Ethernet Hub or Switch
2	Cyclades ACS 5000 Console Server	**7**	Remote Data Server
3	Power Cable	**8**	Ethernet Router
4	Ethernet CAT 5 Cable	**9**	Local User
5	TACACS Server	**10**	User

To test a CAS configuration:

1. Create a new user in the local database.

   # **adduser <username>**

```
# passwd <username>
```

2. Make sure the physical connection between the console server and the servers is correct.
3. Confirm the communication parameters (9600 bps, 8N1) are set on both the server and the console server.
4. Make sure the server is configured to route console data to its serial console port (Console Redirection).
5. Telnet to the server connected to Port 1.
6. From a server on the local network (not from the console), try to Telnet to the server connected to the port 1 of the console server using the following command.

   ```
   # telnet <ip address> <TCP port>
   ```

7. A Telnet session should open on the server connected to Port 1.
8. To activate and save the changes run the following commands.

   ```
   # runconf
   ```

   ```
   # saveconf
   ```

Terminal Server (TS) profile

The console server provides features for out-of-band management through the configuration of terminal ports. A TS profile allows a terminal user to access a server on the network. The terminal may be either a dumb terminal or a terminal emulation program running on a workstation. Figure 4.3 displays an example of a TS profile. Descriptions follow in Table 4.11.

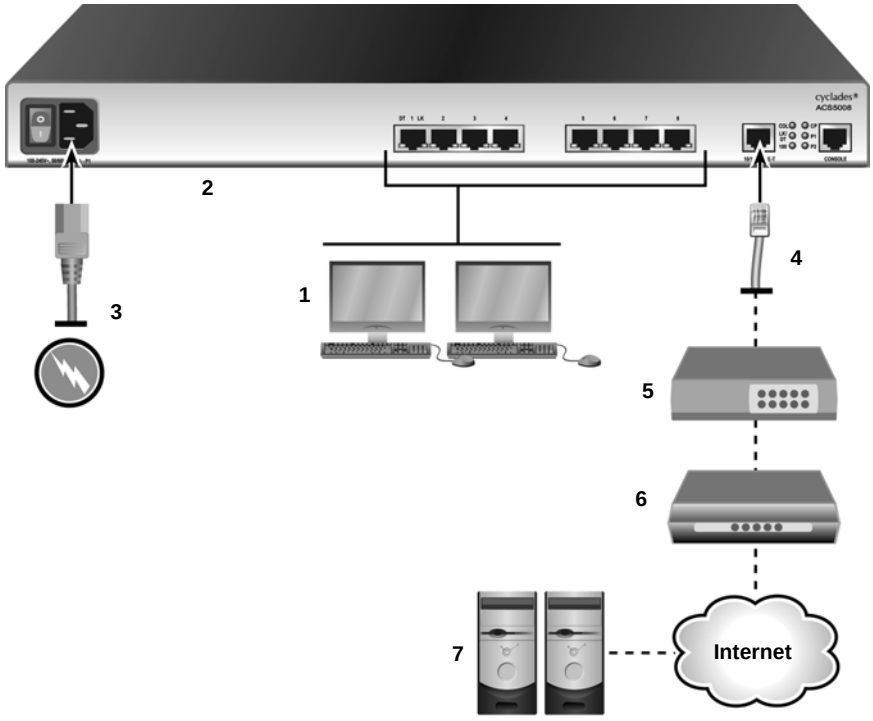

Figure 4.3: Example of TS Configuration Profile

Table 4.11: Example of TS Configuration Profile Descriptions

Item	Description	Item	Description
1	Terminals on Serial Port (dumb terminal or workstation running terminal application)	**5**	Ethernet Hub or Switch
2	Cyclades ACS 5000 Console Server	**6**	Ethernet Router
3	Power Cable	**7**	Remote Servers
4	Ethernet CAT 5 Cable		

To test a TS configuration:

1. Create a new user in the local database.

   # **`adduser <username>`**

   # **`passwd <username>`**

2. Create a new test user and password on the server.
3. From the console, ping the server to make sure it is reachable.

4. Confirm that terminal communication parameters are set to the same as the console server. The console server default communication parameters are at 9600 bps, 8N1.
5. Log in to the server with the newly created username and password.
6. From a terminal connected to the console server, log in to the server using the username and password configured in Step 1.
7. Run the following commands to activate and save your configuration.

   # **runconf**

   # **saveconf**

Dial-in access profile

The console server serial ports may be configured to allow remote users to access the local network through a modem.

To configure a dial-in access profile:

1. Configure the serial port for PPP protocol.
2. Create a new user on the authentication server.
3. From the console, ping the authentication server to make sure it is reachable.
4. Confirm modem settings. The console server is set for communication at 57600 bps, 8N1. The modems should be programmed to operate at the same speed and the same flow control on the DTE interface.
5. Make sure the server is configured to route console data to the serial console port.
6. Dial-in to the console server from a remote server using the username and password created. The server dialing in must be configured to receive its IP address from the remote access server (the console server in this case) and to use PAP authentication.
7. Run the following command to activate and save your configuration.

   # **runconf**

   # **saveconf**

Figure 4.4 displays an example of a dial-in access profile with Radius authentication and ppp protocol on the serial lines. Descriptions follow in Table 4.12.

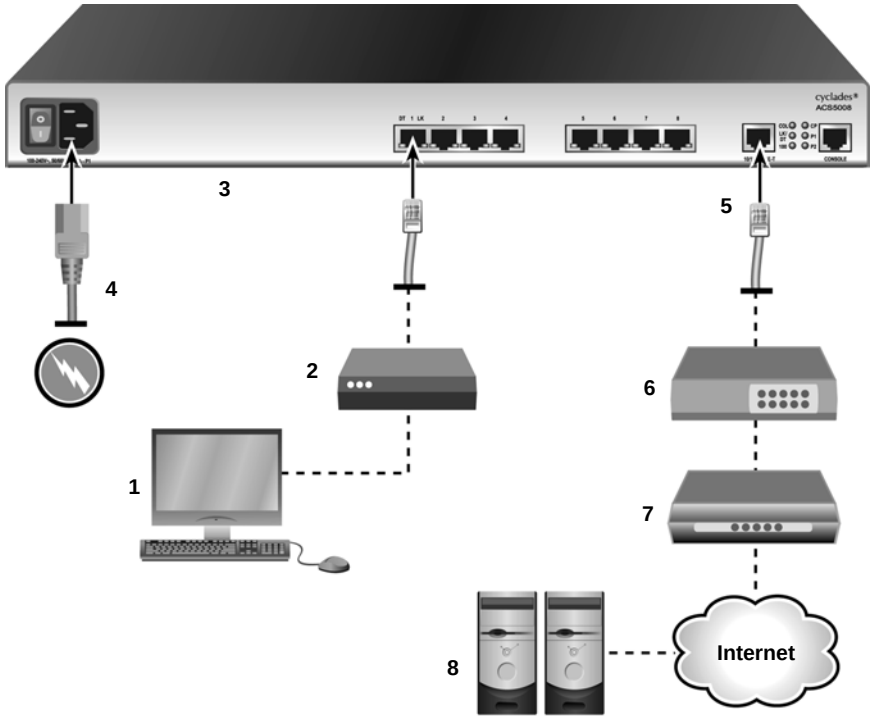

Figure 4.4: Example of Dial-in Access Profile

Table 4.12: Example of Dial-in Access Profile Descriptions

Item	Description	Item	Description
1	Dial-up Management Station	**5**	Ethernet CAT 5 Cable
2	Modem	**6**	Ethernet Hub or Switch
3	Cyclades ACS 5000 Console Server	**7**	Ethernet Router
4	Power Cable	**8**	Remote Servers

CHAPTER 5

Administration

Process Monitoring

The command “w” displays information about the currently logged-in users and their processes. It calls two commands, w_ori and w_cas. The w_ori is the new name of the original command -w and the w_cas displays the CAS session’s information.

The header of w_ori displays the current time, how long the system has been running, how many users are currently logged on (excluding the CAS users) and the system load averages for the past one, five and fifteen minutes.

The following entries are displayed for each user (excluding the CAS users).

- Login name
- The tty name
- The remote host
- Login time
- Idle time
- JCPU time - It is the time used by all processes attached to the tty
- PCPU time - It is the time used by the current process named in the “what” field
- The command line of the user’s current process

The header of w_cas displays how many CAS users are currently logged on. The following entries are displayed for each CAS user.

- Login name
- The tty name
- The remote host and remote port
- Login time
- The process ID
- The command line of the current process

The Process Table

The process table displays which processes are running. Type **ps -a** to see a table similar to the following.

Table 5.1: Process Table

PID	UID	VmSize	State	Command
1	root	592	S	/sbin/inetd
31	root	928	S	/sbin/inetd
32	root	584	S	/sbin/cy_ras
36	root	1148	S	/sbin/cy_wdt_led wdt led
154	root	808	R	/ps -a

To restart the cy_ras process, use its process ID or execute the following command.

```
# runconf
```

This executes the ps command, searches for the cy_ras process id, then sends the signal hup to the process, all in one step. Never kill cy_ras with the signals -9 or SIGKILL.

Start and Stop Services

This feature enables or disables services without rebooting the console server.

Syntax

```
daemon.sh [-h|help] | [{[stop|restart] service_id}...]
where service_id may be any choice of:
EVTGEN NIS RPC DB NET LOG SSH NTP SNMP IPSEC PMD LP WEB LOGPIPE ADSAP2
```

The daemon.sh may be executed in two ways.

1. Without parameters in the command line. It checks the configuration files of the service and restart or stop it if needed.
2. It performs the requested action (stop/restart) in the list of services given in the command line regardless of any configuration changes.

The following example restarts power management and data buffering services and it stops SSH and network timer client services.

```
# daemon.sh PMD stop SSH NTP restart DB
```

Syslog-ng

Syslog-ng daemon reads log system console messages and log files on remote syslog servers as specified by its configuration file. In addition, syslog-ng may filter messages based on its content and perform an action, for example send an email or pager message. The /etc/syslog-ng/syslog-ng.conf file is used to perform specific configurations.

To configure syslog-ng:

1. Define Global Options

```
options { opt1(params); opt2(params); ... };
```

Table 5.2: Global Options Parameters (Syslog-ng Configuration)

Option	Description
time_reopen(n)	The time to wait before a dead connection is re-established.
time_reap(n)	The time to wait before an idle destination file is closed.
sync_freq(n)	The number of lines buffered before written to file. (The file is synced when this number of messages has been written to it.)
mark_freq(n)	The number of seconds between two MARKS lines.
log_fifo_size(n)	The number of lines fitting to the output queue.
chain_hostname (yes/no) or long_hostname (yes/no)	Enable/disable the chained hostname format.
use_time_recvd (yes/no)	Use the time a message is received instead of the one specified in the message.
use_dns (yes/no)	Enable or disable DNS usage. syslog-ng blocks on DNS queries, so enabling DNS may lead to a Denial of Service attack.
gc_idle_threshold(n)	Sets the threshold value for the garbage collector, when syslog-ng is idle. GC phase starts when the number of allocated objects reach this number. Default: 100.
gc_busy_threshold(n)	Sets the threshold value for the garbage collector. When syslog-ng is busy, GC phase starts.
create_dirs(yes/no)	Enable the creation of new directories.
owner(name)	Set the owner of the created file to the one specified. Default: root.
group(name)	Set the group of the created file to the one specified. Default: root.
perm(mask)	Set the permission mask of the created file to the one specified. Default: 0600.

2. Define Sources

```
source <identifier> { source-driver([params]); source
driver([params]); ...};
```

where,

- identifier - Uniquely identifies a given source.
- source-driver - A method of retrieving a given message.
- params - Each source-driver takes a required or an optional parameter.

Table 5.3: Source Drivers Parameters (Syslog-ng Configuration)

Option	Description
internal()	Messages are generated internally in syslog-ng.
unix-stream (filename [options]) and unix-dgram (filename [options])	They open the given AF_UNIX socket and start listening for messages. Options: owner(name), group(name), perm(mask) are equal global options. keep-alive(yes/no) - Selects whether to keep connections opened when syslog-ng is restarted. May be used only with unix_stream. Default: yes max-connections(n) - Limits the number of simultaneously opened connections. May be used only with unix_stream. Default: 10.
tcp([options]) and udp([options])	These drivers let you receive messages from the network, and as the name of the drivers show, you may use both TCP and UDP. None of tcp() and udp() drivers require positional parameters. By default they bind to 0.0.0.0:514, which means that syslog-ng listens on all available interfaces. Options: ip(<ip address>) - The binding IP address. Default: 0.0.0.0. port(<number>) - UDP/TCP port used to listen messages. Default: 514. max-connections(n) - Limits the number of simultaneously opened connections. Default: 10.
file(filename)	Opens the specified file and reads messages.
pipe(filename)	Opens a named pipe with the specified name and listens for messages. (You need to create the pipe using the mkfifo command).

The following are examples of how to define sources.

- Read from a file.

```
source <identifier> {file(filename);};
```

- Read messages from /temp/file1 file.

```
source file1 {file('/temp/file1');};
```

- Receive messages from the kernel.

```
source s_kernel { file('/proc/kmsg'); };
```

- Receive messages from local syslogd clients.

```
source sysl {unix-stream('/dev/log');};
```

- Receive messages from remote syslogd clients.

```
source s_udp { udp(ip(<cliente ip>) port(<udp port>)); };
```

- Listen to messages from all machines on UDP port 514.

```
source s_udp { udp(ip(0.0.0.0) port(514));};
```

- Listen to messages from a client at IP address=10.0.0.1 on UDP port 999.

```
source s_udp_10 { udp(ip(10.0.0.1) port(999)); };
```

3. Define Filters

```
filter <identifier> { expression; };
```

where,

- identifier - Uniquely identifies a given filter.
- expression - Builds a boolean expression using internal functions.

Table 5.4: Filters Parameters (Syslog-ng Configuration)

Option	Description
facility (<facility code>)	Selects messages based on their facility code.
level(<level code>) or priority (<level code>)	Selects messages based on their priority.
program(<string>)	Tries to match the <string> to the program name field of the log message.
host(<string>)	Tries to match the <string> to the hostname field of the log message.
match(<string>)	Tries to match the <string> to the message itself.

The following are examples of how to define filters.

- To filter by facility.

```
filter f_facilty { facility(<facility name>); };
```

Examples:

```
filter f_daemon { facility(daemon); };
filter f_kern { facility(kern); };
filter f_debug { not facility(auth, authpriv, news, mail); };
```

- To filter by level.

  ```
  filter f_level { level(<level name>);};
  ```

  Examples:

  ```
  filter f_messages { level(info .. warn)};
  filter f_emergency { level(emerg); };
  filter f_alert { level(alert); };
  ```

- To filter by matching a string in the received message.

  ```
  filter f_match { match(‘string’); };
  ```

  Example to filter by matching the string named:

  ```
  filter f_named { match(‘named’); };
  ```

- To filter alarm messages.

  ```
  filter f_alarm { facility(local[0+<conf.DB_facility>]) and level(info)
  and match('ALARM') and match('<your string>'); } ;
  ```

  Example to filter alarm message with the string kernel panic:

  ```
  filter f_kpanic { facility(local[0+<conf.DB_facility>]) and
  level(info) and match('ALARM') and match('kernel panic'); };
  ```

- To eliminate SSHD debug messages.

  ```
  filter f_sshd_debug { not program('sshd') or not level(debug); };
  ```

- To filter the syslog_buffering.

  ```
  filter f_syslog_buf { facility(local[0+<conf.DB_facility>]) and
  level(notice); };
  ```

- To define actions (destinations).

  ```
  destination <identifier> {destination-driver([params]); destination-
  driver([param]);..};
  ```

  where,

  - identifier - Uniquely identifies a given destination.
  - destination driver - Configures a method of output for a given message.
  - params - Configures a required or an optional parameter for each destination-driver.

Table 5.5: Destination Drivers Parameters (Syslog-ng Configuration)

Option	Description
file (filename[options])	This is one of the most important destination drivers in syslog-ng. It allows you to output log messages to the named file. The destination filename may include macros (by prefixing the macro name with a '$' sign) which gets expanded when the message is written. Since the state of each created file must be tracked by syslog-ng, it consumes some memory for each file. If no new messages are written to a file within 60 seconds (controlled by the time_reap global option), it's closed, and its state is freed. Available macros in filename expansion: HOST - The name of the source host from where the message originated. FACILITY - The name of the facility from which the message is tagged. PRIORITY or LEVEL - The priority of the message. PROGRAM - The name of the program the message was sent by. YEAR, MONTH, DAY, HOUR, MIN, SEC - The year, month, day, hour, min, sec of the message was sent. TAG - Equals FACILITY/LEVEL. FULLHOST - The name of the source host and the source-driver: <source-driver>@<hostname> MSG or MESSAGE - The message received. FULLDATE - The date of the message was sent. Available options: log_fifo_size(number) - The number of entries in the output file. sync_freq(number) - The file is synced when this number of messages has been written to it. owner(name), group(name), perm(mask) - Equals global options. template(“string”) - Syslog-ng writes the “string” in the file. You may use the MACROS in the string. encrypt(yes/no) - Encrypts the resulting file. compress(yes/no) - Compresses the resulting file using zlib.
pipe (filename[options])	This driver sends messages to a named pipe. Available options: owner(name), group(name), perm(mask) - Equals global options. template(“string”) - Syslog-ng writes the “string” in the file. You may use the MACROS in the string.
unix-stream(filename) and unix-dgram(filename)	This driver sends messages to a UNIX socket in either SOCKET_STREAM or SOCK_DGRAM mode.
udp(“<ip address>” port(number);) and tcp("<ip address>" port(number);)	This driver sends messages to another host (ip address/port) using either UDP or TCP protocol.
program(<program name and arguments>)	This driver fork executes the given program with arguments and sends messages down to the stdin of the child.
usertty(<username>)	This driver writes messages to the terminal of a logged-in username.

The following is an example of how to send an email.

```
destination <ident> { pipe('/dev/cyc_alarm' template('sendmail
<pars>'));};
```

where <ident> uniquely identifies the destination.

Table 5.6: Send Email Parameters

Email field	Parameter
To address	-t <name>[,<name>]
CC address	[-c <name>[,<name>]]
Bcc address	[-b <name>[,<name>]]
Reply-to address	[-r <name>[,<name>]]
From address	-f <name>
Subject	-s \"<text>\"
Message	-m \"<text message>\"
SMTP server	-h <IP address or name>
Port used. default:25	[-p <port>]

Table 5.7 shows the message mount parameters.

Table 5.7: Message Mount Parameters

Parameter	Description
$FULLDATE	The complete date when the message was sent.
$FACILITY	The facility of the message.
$PRIORITY or $LEVEL	The priority of the message.
$PROGRAM	The message was sent by this program (BUFFERING or SOCK).
$HOST	The name of the source host.
$FULLHOST	The name of the source host and the source driver Format: <source>@<hostname>
$MSG or $MESSAGE	The message received.

The following example displays an email sent to z@none.com (SMTP's IP address 10.0.0.2) from the email address a@none.com with subject “ALARM”. The message carries the current date, hostname of the console server and the message received from the source.

```
destination d_mail1 {
     pipe('/dev/cyc_alarm'
        template('sendmail -t z@none.com -f a@none.com -s \"ALARM\" \\
           -m \'$FULLDATE $HOST $MSG\' -h 10.0.0.2'));
};
```

The following example shows how to send a message to the sms server.

```
destination <ident> {pipe(‘/dev/cyc_alarm’ template(‘sendsms
<pars>’));};
```

where <ident> uniquely identify the destination. The parameters are:

```
pars: -d <mobile phone number>
-m \’<message - max.size 160 characters>\’
-u <username to login on sms server>
-p <port sms - default : 6701>
<server IP address or name>
```

The following example sends a page to phone number 123 (Pager server at 10.0.0.1) with the message carrying the current date, the hostname of the console server and the message received from the source.

```
destination d_pager {
     pipe(‘/dev/cyc_alarm’
     template(‘sendsms -d 123 -m \’$FULLDATE $HOST $MSG\’
10.0.0.1’));
};
```

Sending an snmptrap

```
destination <ident> {pipe(‘/dev/cyc_alarm’ template(‘snmptrap
<pars>’)); };
```

where <ident> uniquely identify the destination. The parameters are:

- pars : -v 1
- <snmptrapd IP address>
- -c public : community
- \"\" : enterprise-oid
- \"\" : agent/hostname
- <trap number> : 2-Link Down, 3-Link Up, 4-Authentication Failure

- 0 : specific trap
- \"\" : host-uptime
- .1.3.6.1.2.1.2.2.1.2.1 :interfaces.iftable.ifentry.ifdescr.1
- s : the type of the next field (it is a string)
- \"<message - max. size 250 characters>\"

The following example sends a Link Down trap to a server at 10.0.0.1 with message carrying the current date, the hostname console server and the message received from the source.

```
destination d_trap {
pipe("/dev/cyc_alarm"
template("snmptrap -v 1 -c public 10.0.0.1 public \"\" \"\" 2 0 \"\"
\\ .1.3.6.1.2.1.2.2.1.2.1 s \"$FULLDATE $HOST $MSG\" "));
};
```

Sending a message to a remote syslogd server

```
destination d_udp { udp("<remote IP address>" port(514)); };
```

The following example sends syslogs to syslogd located at 10.0.0.1 :

```
destination d_udp1 { udp("10.0.0.1" port(514)); };
```

Connecting sources, filters and actions

To connect the sources, filters and actions use the following statement. An action is an incoming message from one of the listed sources. A match for each of the filters is sent to the listed destination.

```
log { source(S1); source(S2); ...
filter(F1);filter(F2);...
destination(D1); destination(D2);...
};
```

where,

- Sx - Identifies the defined sources.
- Fx - Identifies the defined filters.
- Dx - Identifies the defined actions or destinations.

Examples of connecting sources, filters and actions:

- To send all messages received from local syslog clients to console.

```
log { source(sysl); destination(d_console);};
```

- To write all messages with levels info, notice or warning and received from syslog clients (local and remote) to /var/log/messages file.

```
log { source(sysl); source(s_udp); filter(f_messages);
destination(d_messages); };
```

- To send an email if message received from local syslog client has the string kernel panic.

```
log { source(sysl); filter(f_kpanic); destination(d_mail1); };
```

- To send an email and pager if message received from local syslog client has the string "root" login.

```
log { source(sysl); filter(f_root); destination(d_mail1);
destination(d_pager); };
```

- To send messages with facility kernel and received messages from syslog clients (local and remote) to remote syslogd.

```
log { source(sysl); source(s_udp); filter(f_kern); destination(d-
udp1); };
```

To use syslog-ng configuration with syslog buffering feature:

This configuration example uses the syslog buffering feature and sends messages to the remote syslogd (10.0.0.1).

1. In /etc/portslave/pslave.conf file configure the syslog buffering parameters.

```
conf.DB_facility 1
all.syslog_buffering 100
```

2. Add the following lines to /etc/syslog-ng/syslog-ng.conf file.

```
#local syslog clients
source src { unix-stream("/dev/log"); };
destination d_buffering { udp("10.0.0.1"); };

filter f_buffering { facility(local1) and level(notice); };
#send only syslog_buffering messages to remote server
log { source(src); filter(f_buffering);
destination(d_buffering); };
```

To configure Syslog-ng with multiple remote syslog servers:

1. Configure syslog facility number to receive messages. The remote syslog server filters receive messages according to this parameter.

```
cli> config network syslog facility <local0-local7>
```

2. Configure the server's IP address where syslog messages are sent. Repeat this step to add additional remote servers.

```
cli> config network syslog add server <ip address>
```

3. Activate and save your configuration.

Syslog Messages

The console server may generate syslog messages to enable system administrators to monitor system changes. Syslog messages are generated when specific actions are performed or certain conditions are met through user entered commands. The system generates and sends messages to a syslog server using the following format.

- Tag - a fixed string used by the user to create filters

  `EVT[<event number>]:`
- Text - the text that contains the condition or action

You may use the information provided in the CYCLADES-ACS5000-TRAP-MIB.ASN to create filters and generate alarms about the console server events.

DCD ON/OFF Syslog Messages

The console server may generate an alert when a serial console cable is removed from the console server or when the serially attached server is turned off. Also, when a modem is connected, this feature may detect if the modem is still turned on and active.

The DCD signal is monitored and a syslog message is generated when the state of the signal changes. The syslog message may be handled by syslog-ng to generate an event notification.

To configure DCD syslog messages:

1. Open the /etc/portslave/pslave.conf file.

   `# vi /etc/portslave/pslave.conf`
2. Set the all.dcd or sXX.dcd parameter to 1 in the /etc/portslave/pslave.conf file.

   `all.dcd 1`

   -or-

   `sXX.dcd 1`

   Where XX is the desired port number.
3. Configure the event_notif conf file to monitor DCD status.

   The following example displays generating syslog messages if the DCD signal changes its state.

   `9=2`

   `10=2`
4. Save the configuration.

   `# saveconf`

Notifications and Alarms

System notifications allow an administrator to manage servers by filtering the messages generated from a server's console port. It helps with sending email or pager notifications based on the server's message content.

Configuring alarm notification

```
cli> config administration notifications [parameter] <value>
```

Table 5.8: System Notifications Parameters

Parameter Level1	Parameter Level2	Value	Description
addemail		<trigger string>	Sends a message to the configured email address if the defined string appears.
	add Email>		
	from		
	to		
	subject		
	body		
	smtpserver		
	smtpport		
	cancel		
addpager		<trigger string>	Sends a message to the configured pager if the defined string appears.
	add Pager>		
	number	<string>	
	smsport	<number>	
	smsserver	<string>	
	smsusername	<string>	
	text	<string>	
	cancel		

Table 5.8: System Notifications Parameters (Continued)

Parameter Level1	Parameter Level2	Value	Description
addsnmptrap		<trigger string>	Sends a SNMP trap to the configured server if the defined string appears.
	add Snmptrap>		
	body	<string>	
	community	<string>	
	oid	<string>	
	server	<string>	
	cancel		
	trapnum	0-6 authfailure coldstar egpneighborloss enterprisespecific linkdown linkup warmstart	
alarm		yes|no	Activate or deactivate the alarm feature. If you don't enable it, syslog messages won't be generated when there is incoming data from the ports.
delete			Delete any previously configured string.
edit			Edit any previously configured string.

To configure notifications:

The following example demonstrates configuring the console server to send an email every time the root user logs into a server connected to a serial port. The trigger string is configured as root login. The server connected to the console server must be properly configured to send Syslog messages.

1. Enable alarm notification, otherwise messages received through the serial ports are ignored.

   ```
   cli> config administration notifications alarm yes
   ```

2. Add a trigger string.

   ```
   cli> config administration notifications addemail "root login"
   ```

3. Configure the email notification parameters and SMTP server and port id.

   ```
   add Email>body "Testing configuration"
   add Email>from ACSConsoleServer
   add Email>to someone@yourdomain.com
   add Email>smtpserver 200.200.200.2
   ```

```
add Email>smtpport 25
add Email>subject "Testing Config"
```

4. Activate and save your configuration.

Dual Power Management

The console server comes with two power supplies which it may self-monitor. If either of them fails, two actions are performed, sounding a buzzer and generating a syslog message. This automanagement may be disabled (no actions are taken) or enabled (default), any time by issuing the commands.

```
# signal_ras buzzer off
```

```
# signal_ras buzzer on
```

To disable the buzzer in boot time, edit the shell script /bin/ex_wdt_led.sh and remove the keyword buzzer. The buzzer won't sound if there is a power failure in any power supply. This parameter does not affect the behavior of the command signal_ras buzzer on/off. To make this change effective even after future reboots, create a line with /bin/ex_wdt_led.sh in /etc/config_files, save and quit the file and run saveconf.

NOTE: This section applies only to the dual power supply model of the console server.

Date and Time, Timezone and Daylight Saving

To adjust the date and time, use the date command. Timezone is configured using the CLI utility or web manager (see the ACS 5000 Installation/Administation/User Guide for using the web manager to set time, date and timezone information.)

NOTE: Setting the system timezone creates a new file called /etc/localtime, which erases /etc/TIMEZONE.

Daylight Saving Time (DST)

When the DST parameter is set to on, the console server automatically adjusts its time information to comply with the time shift appropriate to the target timezone. For states, countries or regions that do not observe daylight saving time, the dst parameter must be set to off even if other regions in the target timezone do observe the daylight saving time change.

In rare occurrences or under special circumstances, a region or country might require that a customized daylight saving time be used. Such circumstances might require a temporary or permanent change of date for the beginning and ending of daylight time, or a time offset greater or less than the usual one hour. Instructions follow for customizing the daylight saving time parameters.

Enter the following command to set the date and time. For configuration parameters see Table 5.9.

```
cli> config administration date/time [parameter] <value>
```

Table 5.9: Date and Time Configuration Parameters

Parameter	Value
date	<mm/dd/yy>
time	<hh:mm:ss>

Enter the following command to set the timezone. For configuration parameters see Table 5.10.

```
cli> config administration timezone [parameter] <value>
```

Table 5.10: Timezone Configuration Parameters

Parameter Level1	Parameter Level2	Value	Description
Custom	zonelabel	Timezone name	May be any custom name you choose (such as, “London,” “ChicagoOffice,” or “Sydney”), or may be a numerical value.
	acronym	Timezone acronym	The abbreviated name for the zonelabel. For example “PST” for “Pacific Standard Time.”
	gmtoff	<hh:mm>	GMT Offset: This is the number of hours either ahead (+) or behind (-) Greenwich Mean Time (GMT) in hours. For example, PST, the offset is -8:00 hours.
	dst	off|on	Daylight Saving Time (DST): Set to “on” for custom daylight saving time settings to be active.
	dstacronym	DST acronym	The abbreviated name used to describe the timezone when daylight saving time is in effect. For example, “PDT” for Pacific Daylight Time.
	dstsave	<hh:mm>	This is the amount of time that the clock moves forward or back at the beginning and end of daylight saving time for the target timezone.
	dststartday	(see format in Description)	The day <Jan, . . . ,Dec>/<1st,...,4th,last>/<Sun,...,Sat>/ that DST starts for the target timezone.
	dststarttime	<hh:mm>	The precise time of day (hh:mm) that DST starts for the target timezone.
	dstendday	(see format in Description)	The day <Jan, . . . ,Dec>/<1st,...,4th,last>/<Sun,...,Sat>/ that DST ends for the target timezone.
	dstendtime	<hh:mm>	The precise time of day (hh:mm) that DST ends for the target timezone.

Table 5.10: Timezone Configuration Parameters (Continued)

Parameter Level1	Parameter Level2	Value	Description
Standard		01h_east_GMT . . . 14h_east_GMT GMT 01h_west_GMT . . . 12h_west_GMT	

The following are examples of configuring timezones.

```
cli> config administration timezone custom acronym PDT
cli> config runconfig
cli> config savetoflash

cli> config administration timezone standard 08h_west_GMT
cli> config runconfig
cli> config savetoflash

cli> config administration timezone custom dst on dstacronym PDT
dststartday Mar/2nd/Sun dstendday Nov/1st/Sun
cli> config runconfig
cli> config savetoflash
```

Network Time Protocol (NTP)

The ntpclient is a Network Time Protocol client for UNIX and Linux based systems. In order for the console server to work as an NTP client, the IP address of the NTP server must be configured.

To configure an NTP server:

1. Execute the following command to configure the NTP server IP address.

   ```
   cli> config administration ntp <NTP server IP address>
   ```

2. Activate and save your configuration.

NOTE: To deactivate the NTP service you need to configure date by issuing the following command:
cli> config administration date/time date <mm/dd/yyyy>

Session Sniffing

When multiple sessions are allowed for one serial port, the behavior of the console server is as follows.

- The first user to connect to the port opens a common session.
- From the second connection onwards, only admin users are allowed to connect to that port. The console server opens the following menu to these administrators, which is defined by the parameter all.admin_users or sN.admin_users in the file pslave.conf.

```
_____________________________________________________
* * * ttySN is being used by (<first_user_name>) !!!
*
1 - Initiate a regular session
2 - Initiate a sniff session
3 - Send messages to another user
4 - Kill session(s)
5 - Quit

Enter your option:
_____________________________________________________
```

If you select 1 - Initiate a regular session, the serial port is shared with the users that were previously connected. You are able to read and write to the serial port.

If you select 2 - Initiate a sniff session, you may read everything that is sent or received through the serial port, according to the parameter all.sniff_mode or sN.sniff_mode.

If you select 3 - Send messages to another user, the console server sends your messages to all the sessions, but not to the tty port. Everyone connected to that port sees all exchanges of information as if they were physically in front of the console. These messages are formatted as,

```
[Message from user/PID] <<message text goes here>> by the ACS.
```

If you select 4 - Kill session(s), the console server displays a list of PID/username pairs. You are able to select a session by typing its PID, or all to kill all sessions. If you kill all the regular sessions, a regular user session initiates automatically.

Select Option 5 - Quit to close the current session and the TCP connection.

NOTE: Typing all.escape_char or sN.escape_char from the sniff session or send message mode makes the console server to show the previous menu. If you kill all regular sessions using option 4, your session initiates as a regular session automatically.

To configure session sniffing:

1. Execute the following command for one or multiple serial ports. Refer to Table 5.11 for session sniffing parameters.

   ```
   cli> config physicalports <'all' or range/list[1-xx]> multiuser
   [parameter] <value>
   ```

2. Activate and save your configuration.

Table 5.11: Session Sniffing Parameters

Parameter	Value	Description
hotkey	<^(character)>	To configure the escape character. The selected character must be preceded by the '^' character. For example, ^k.
notifyusers	yes|no	To configure multiuser notification.
multisessions	no | ro | rw | yes	To configure multiple sessions.
privilegeusers	<list of user names separated by commas>	To determine which users may receive the sniff menu.
sniffmode	in | inout | no | urt	Determines what other users connected to the very same port may see of the session of the first connected user (main session). Valid values are: *in* - shows data written to the port; *out* - shows data received from the port; *in/out* - shows both streams; *off* - disables sniffing.

Data Buffering

Data buffering may be done in local files or in remote files through NFS. When using remote files, the limitation is imposed by the remote server (disk/partition space) and the data is kept in linear (sequential) files in the remote server. When using local files, the limitation is imposed by the size of the available ramdisk. You may wish to have data buffering done in file, syslog or both. For syslog, all.syslog_buffering and conf.DB_facility are the parameters to be dealt with, and syslog-ng.conf file should be set accordingly. Please see *Syslog-ng* on page 71 for the syslog-ng configuration file. The data buffering parameters are configured in file all.data_buffering.

Conf.nfs_data_buffering is a remote network file system where data buffering is written to, instead of the default directory /var/run/DB. When commented, it indicates local data buffering. The directory tree to which the file is written must be NFS-mounted and the local path should point to/mnt/DB_nfs. The remote host must have NFS installed and the administrator must create, export and allow read/write privileges to the directory. The size of this file is not limited by the value of

the parameter s1.data_buffering, though the value cannot be zero since a zero value turns off data buffering.

The conf.nfs_data_buffering parameter format is,

```
<server name or IP address>:<remote pathname>
```

For example, if data buffering is enabled for port 1, the data is stored in the file ttyS1.data in local directory /var/run/DB or in remote server indicated by conf.nfs_data_buffering.

Ramdisk

Data buffering files are created in the directory /var/run/DB. If the parameter s<nn>.alias is configured for the port <nn>, this name is used. For example, if the alias is called fremont_server, the data buffering file is named fremont_server.data.

Linear vs. Circular buffering

For local data buffering, this parameter allows users to buffer data in either a circular or linear fashion. Circular format (cir) is a revolving buffer file that is overwritten whenever the limit of the buffer size (set by all.data_buffering) is reached. In linear format (lin), data transmission between the remote device and the serial port ceases once the 4k bytes Rx buffer in the kernel is reached. Then if a session is established to the serial port, the data in the buffer is shown (all.dont_show_DBmenu or sxx.dont_show_DBmenu must be 2), cleared and data transmission is resumed. Linear buffering is impossible if flow control is set to none. Default is cir.

To configure data buffering:

1. Execute the following command to configure data buffering. Refer to Table 5.12 for the configuration parameters.

   ```
   cli> config physicalports all databuffering [parameter] <value>
   ```

2. Activate and save your configuration.

Table 5.12: Data Buffering Configuration Parameters

Parameter	Value	Description
bufferonlynosession	no|yes	Buffer only when connected to the serial port.
buffersyslogonlynosession	no|yes	Only syslog is buffered.
desttype	local | remote	Define the data buffering location.
filesize	<file size in bytes>	Defines the maximum size of the data buffer file. This parameter must be greater than zero otherwise all parameters relating to data buffering are disregarded.
mode	cir | lin	Choose between circular or linear data buffering.
nfspath	<pathname>	Define the NFS path.

Table 5.12: Data Buffering Configuration Parameters (Continued)

Parameter	Value	Description
showmenu	file | fileanderase | no | noerase | yes	Control the DB menu options.
syslogserver	<n.n.n.n>	Defines the IP address of the Syslog server.
syslogsize	<record length in bytes[40-255]>	Maximum size of syslog data buffer message.
syslogfacility	local0 - local7	Defines the facility number for messages generated by the console server to be sent to the Syslog server.
timestamp	no|yes	Choose YES to enable timestamp and NO to disable it.

Menu Shell

This application allows you to customize a menu presented to users when they connect to the console server from a terminal. The menu may be configured to allow users to connect to different servers on the local network.

When the menu shell is configured you may connect to the console server using a serial terminal. You may select a server to connect to from the list or exit the system.

```
Welcome!
1) Sun server
2) Dell server
3) Linux server
4) Quit
Option ==>
```

To configure and set up a Menu Shell:

1. Assign the menu shell to users who require access using the options provided by the menush_cfg utility.

Type menush_cfg and use the available options to define a menu title and menu commands.

```
-------------------------------------------------
MenuShell Configuration Utility
-------------------------------------------------
Please choose from one of the following options:
1. Define Menu Title
2. Add Menu Option
3. Delete Menu Option
4. List Current Menu Settings
5. Save Configuration to Flash
6. Quit
Option ==>
```

2. Choose Add Menu Option and complete the requested fields. The following question defines the action that must be taken.

   ```
   Enter the command for the new menu option:
   ```

3. Select option 5 to save the configuration changes to Flash.

NOTE: Action may be telnet host_ip or ssh -l username host_ip where host_ip is the IP address of the target server.

To assign ports to the menu shell:

1. If no authentication is required to gain access to the menu, configure the following parameters in /etc/portslave/pslave.conf for the ports that use this menu shell.

   ```
   s<x>.protocol telnet
   conf.telnet /bin/menush
   s<x>.authtype none
   ```

   Where <x> is the port number being configured.

2. If authentication is required to gain access to the menu, the user's default shell must be modified to run the /bin/menush. In /etc/passwd the shell should be changed as per the following example.

   ```
   user:FrE6QU:505:505:Embedix User,,,:/home/user:/bin/menush
   ```

3. In pslave.conf, the port where the serial terminal is attached must be configured for login with local authentication.

   ```
   s<x>.protocol login
   s<x>.authtype local
   ```

Where <x> is the port number being configured.

4. Activate and save the configuration changes.

   # **runconf**

   # **saveconf**

To set up which servers the users may access:

1. Enter the following command to set up a menu, which is prompted when you connect from a dumb terminal. Refer to Table 5.13 for configuration parameters.

   ```
   cli> config applications terminalmenu add [parameter] <value>
   ```

2. Activate and save your configuration.

Table 5.13: Terminal Profile Menu Configuration Parameters

Parameter	Value	Description
menutitle	<string>	Type a description for the menu title bar.
actionname	<action name>	Enter n identification for the command. For example, server name.
command	<string>	Enter a command such as telnet host_ip.

NOTE: You may open an SSH connection to the desired server. To do so, substitute telnet host_ip with ssh -l username host_ip.

Terminal Appearance

You may change the banner appearance when a connection is made to a server. The banner appearance may be port-specific or a unified banner for all ports.

To configure the terminal appearance:

1. Enter the following command to configure a banner for one or multiple serial ports.

   ```
   cli> config physicalports [all|range] other banner "<login banner>"
   ```

NOTE: A banner string with spaces must be enclosed by double quotes "<string1 string2>".

2. Activate and save your configuration.

SUDO Configuration Group

SUDO configuration group allows users belonging to the administrator (admin) group, by way of commands from the shell command line, to configure the console server's features provided by the web manager and CLI.

NOTE: As supplied, the console server (version 3.3 and greater) provides a user admin from the admin group with the password avocent. The username admin cannot be added or deleted from the web manager, or the CLI, so if a user with the username admin belonging to the admin group is required, a shell script must be executed by user root from the shell command line.

The sudoers configuration file has already been configured to allow execution and modification of commands, utilities and configuration files by a user from the admin group.

The sudoers file, /etc/sudoers may be edited by user root either to exclude or to include commands, utilities and configuration files that are to be used with the sudo command by users from the admin group.

NOTE: The sudoers file is not saved to Flash automatically. If you make changes to this file and wish to save the changes, follow the standard procedure to save the config_files file.

For an admin group user to be allowed to execute commands from the shell prompt, the sudo command must be used. Commands requiring root access privileges are executed by an admin user with the following command.

```
$ sudo shell_command_|shell_utility|ACS5000_utility [other required parameters]
```

If a user with username admin belonging to the admin group is required, the following shell script must be executed by user root to configure it.

```
# addadmin
```

Saveconf and Restoreconf

The console server has two utilities for saving and restoring the configuration.

Saveconf utility

The saveconf utility automatically creates a file in Flash to save the default and replace flags. The filename is /mnt/flash/config.tgz.You can also save a configuration file to and restore a configuration file from a remote ftp, tftp or ssh server.

Syntax

Enter the following at the shell prompt to see the syntax for the options:

```
# saveconf --help
Usage:
      Save to flash:              saveconf
      Save to storage device:     saveconf sd  [default] [replace]
      Save to local file:         saveconf local <FILE>
      Save to FTP server:        saveconf ftp   <FILE> <FTP_SERVER> <USER>
   <PASSWORD>
      Save to TFTP server:        saveconf tftp  <FILE> <TFTP_SERVER>
      Save to SSH server:        saveconf ssh   <FILE> <SSH_SERVER> <USER>
```

Table 5.14: Saveconf Utility and Storage Device Parameters

Media	Description
<none>	Save the configuration to internal Flash.
local <File>	Save the configuration to the path and filename.
ftp <remote path and filename> <IP address of the FTP server> <username> <password>	Save the configuration to a remote FTP server.
tftp <remote path and filename> <IP address of the TFTP server>	Save the configuration to a remote TFTP server.
ssh <remote path and filename> <IP address of the SSH server> <username>	Save the configuration to a remote SSH server.

Restoreconf utility

Syntax

Enter the following at the shell prompt to see the syntax for the options:

```
# restoreconf --help
Usage:
   Restore from flash :              restoreconf
   Restore from factory default:     restoreconf factory_default
   Restore from local file :         restoreconf local <FILE>
   Restore from FTP server :         restoreconf ftp <FILE> <FTP_SERVER>
   <USER> <PASSWORD>
   Restore from TFTP server :       restoreconf tftp <FILE> <TFTP_SERVER>
```

```
   Restore from SSH server :         restoreconf ssh  <FILE>
<SSH_SERVER> <USER>
```

Table 5.15: Restoreconf Utility and Storage Device Parameters

Media	Description
local <File>	Read the configuration from the path and local file.
local <File>	Read the configuration from the path and filename.
ftp <remote path and filename> <IP address of the FTP server> <username> <password>	Read the configuration from a remote FTP server.
tftp <remote path and filename> <IP address of the TFTP server>	Read the configuration from a remote TFTP server.
ssh <remote path and filename> <IP address of the SSH server> <username>	Read the configuration from a remote SSH server.

Saving or restoring configuration files using CLI

Use the following commands to save or restore configuration files.

- Save to Flash

  `cli>` **`config savetoflash`**

- Save to FTP server

  `cli>` **`administration backupconfig saveto ftpserverip <n.n.n.n> pathname <string> username <string> password <string>`**

- Load from FTP server

  `cli>` **`administration backupconfig loadfrom ftpserverip <n.n.n.n> pathname <string> username <string> password <string>`**

Crond

Crond is a service provided by the console server that allows automatic, periodically-run custom-made scripts. It replaces the need to run commands manually.

The crond daemon configuration is divided in three parts.

- /etc/crontab_files - The name of this file cannot be changed and it must point only to one file.
- Source file - Holds information about frequency of cron jobs and the files that should be executed. It may have any name, since it is pointed out by the /etc/crontab_files.
- Script files - These are the script files that are scheduled and are pointed by the source file explained previously.

The following parameters are created in the /etc/crontab_files file.

- Status - Active or inactive. The script does not execute if inactive.
- User - The process runs with the privileges of a valid local user.
- Source - Pathname of the crontab file that specifies frequency of execution and the name of shell script. It should be set using the traditional crontab file format.

  Example: active root /etc/tst_cron.src

NOTE: In /etc/crontab_files, you may only have one active entry per user. For instance, from the earlier example, you cannot add another active entry for “root” because it already has an entry. If you want to add more scripts, you may just add them to the source file, for example: (/etc/tst_cron.src).

The /etc/crontab_files file may point to any desired file that calls the scripts to be run. The console server has example file for it (/etc/tst_cron.src). The file that is pointed out in the /etc/crontab_files file must follow the following structure.

```
PATH=/usr/bin:/bin
SHELL=/bin/sh
HOME=/

0-59 * * * * /etc/tst_cron.sh
```

This file is called /etc/tst_cron.src. It can have any name, but it follows structure showed previously. The fourth line of the example file follows this structure: minutes, hours, month day, month, week day and command. It is possible to specify different tasks to run on different dates and times. Each command must be on separated lines. See *Crontab syntax*.

Crontab syntax

A crontab task consists of four date/time fields and a command field. Every minute cron checks all crontabs for a match between the current date/time and their tasks. If there's a match, the command is executed. The system crontab has an additional field User that tells cron with which user id the command should be executed.

The fields are:

- Min - minute of execution, 0-59
- Hour - hour of execution, 0-23
- Mday - day of month of execution, 1-31
- Month - month of execution, 1-12 (or names)
- Wday - day of week of execution, 0-7 (0 or 7 is sunday, or names)
- Command - Anything that may be launched from the command line

Clustering Using Ethernet Interface

Clustering allows cascading multiple console servers so that one master may be used to access all console servers on the network. The master console server can manage up to 1024 serial ports. There are no special connections required between the master and slave console servers, except they all need to be connected in the same physical network. Figure 5.1 displays an example of clustering with one master and two slaves; descriptions follow in Table 5.16.

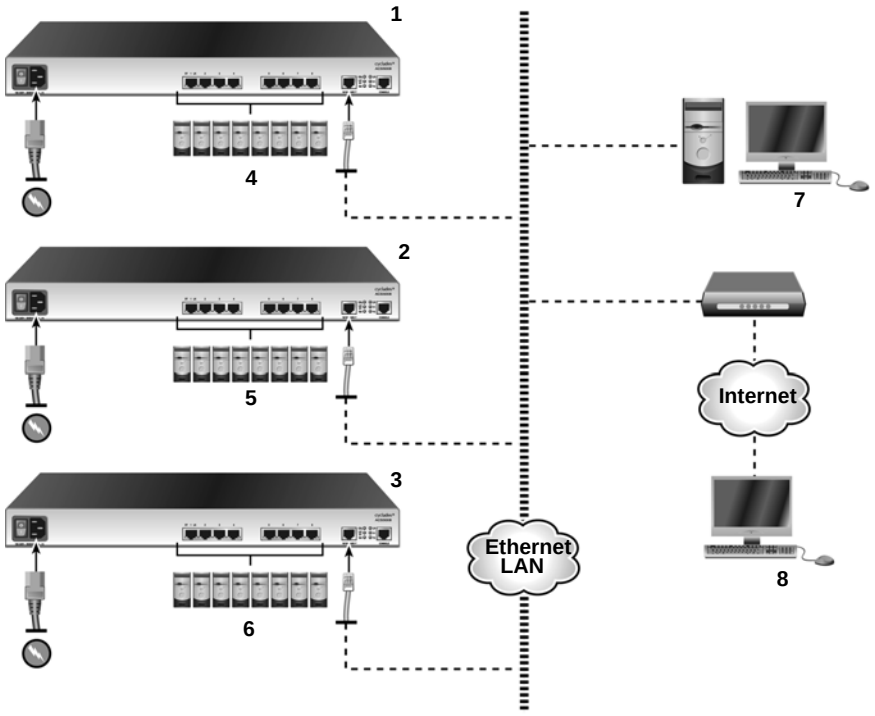

Figure 5.1: An Example on Using the Clustering Feature

Table 5.16: Example of Using the Clustering Feature Descriptions

Item	Description	Item	Description
1	Cyclades ACS 5000 Console Server Master: Ethernet LAN IP: 20.20.20.1 Secondary IP: 209.81.55.110	**5**	Servers on Serial Ports TCP Port Address Range: 7017 - 7032
2	Cyclades ACS 5000 Console Server Master: Ethernet LAN IP: 20.20.20.2	**6**	Servers on Serial Ports TCP Port Address Range: 7033 - 7048

Table 5.16: Example of Using the Clustering Feature Descriptions (Continued)

Item	Description	Item	Description
3	Cyclades ACS 5000 Console Server Master: Ethernet LAN IP: 20.20.20.3	**7**	Management Workstation IP Address: 20.20.20.10
4	Servers on Serial Ports TCP Port Address Range: 7001 - 7016	**8**	Remote Management Workstation

To configure clustering:

1. Execute the following commands to configure slave console servers. Refer to Table 5.17 for configuration parameters.

   ```
   cli> config virtualports addslave <slave ip address>
   cli> Slave [slave ip address]> [parameter] <value>
   ```

2. Activate and save your configuration.

Table 5.17: Clustering Configuration Parameters

Parameter	Value	Description
numports	<list>	Set the total number of ports of the slave unit.
firstlocalportnum	<number[17-1024]>	This parameter act as the numbering continuation in the slave. If the master unit has 16-ports, the first port of the slave unit is the first local port number, which in this case is port 17.
localip	<n.n.n.n>	To set the IP address of the slave.
firstlocaltcpport	<number>	This parameter act as the numbering continuation in the slave. If the master unit has 16-ports, the TCP ports numbers are 7001-7016. In this case the first TCP port number for the slave unit is 7017.
firstremotetcpport	<number>	The first TCP port number in the master unit. In this case is 7001.
protocol	ssh | telnet	Protocol used to access the ports.

Use the following command from the shell prompt to test the configuration.

```
# ssh -l <username>:<TCP port of the serial port in the slave> <IP
address of the master> -p <TCP port of the virtual port in the master>
```

Use the following commands to edit or delete a previously configured virtual port.

```
cli> config virtualports editslave <n.n.n.n>
cli> config virtualports deleteslave <n.n.n.n>
```

CHAPTER 6

Power Management

A Cyclades PM IPDU enables you to remotely control and manage power to target devices attached to the console server. When used in conjunction with the console server, the Cyclades IPDU delivers management capabilities that integrate the console server and power management into a single interface.

In addition to Cyclades PM IPDUs, the following power distribution units are supported by the console server:

- Avocent 1000/2000/3000 Power Management Power Distribution Units (PM PDUs)
- Avocent SPC series power control devices
- Server Technology Sentry™ family of switched cabinet power distribution units (CDUs) and CDU expansion devices
- Server Technology Sentry Power Tower XL™ (PTXL) and Power Tower XM (PTXM) power devices, Server Technology Sentry Smart CDU

NOTE: Configuration and management of Server Technology Sentry Switched and Smart CDUs, PTXL and PTXM models must be handled through the DSView 3 software. The DSView 3 server enables the Server Technology Switched and Smart CDUs, PTXL and PTXM models licensing feature for the selected serial ports in the console server.

The console server may have multiple IPDUs connected to appropriately configured serial ports. Devices may be plugged into outlets on the IPDUs and connected to other serial ports on the console server. In addition, one or more outlets may be configured for each port and controlled individually or simultaneously with other outlets in a configured group. The console server administrator may control all outlets or may assign outlets to individual users or groups of users.

Power Management Protocol

The serial port(s) configured with the Power Management protocol allows you to connect and configure IPDUs using an enabled serial port.

To configure power management protocol:

1. Configure a serial port with the power management protocol.

   ```
   cli> config physicalports <YY> general protocol pm
   ```

2. Configure the connection type SSH, Telnet or both for users to connect to the IPDU.

   ```
   cli> config physicalports <YY> general pmsessions
   <ssh|ssh_telnet|telnet>
   ```

3. Define a unique name for each connected IPDU appliance.

   ```
   cli> config physicalports <YY> general alias <server alias>
   ```

4. Configure the communication protocol for each target server connected to a serial port.

   ```
   cli> config physicalports <XX> general protocol <consoletelnet |
   consolessh | consoletelnetssh>
   ```

   ```
   cli> config physicalports <XX> powermanagement enable
   ```

5. Enable IPDU outlets.

   ```
   enable> outletList <ZZ>
   ```

   For example,

   ```
   enable> outletList ipudA[1], ipduB[2,5-7]
   ```

6. Configure user permission to access an IPDU appliance.

   ```
   enable> pmusers <‘all’ or list of users separated by commas>
   ```

7. Define the hotkey used to open the IPDU menu. The format is **Ctrl (^)** plus a character. The default is **^i**.

   ```
   enable> pmkey ^i
   ```

8. Activate and save your configuration.

   ```
   cli> config runconfig
   ```

   ```
   cli> config savetoflash
   ```

IPDU Configuration and Management

Power management utility

The power management utility may be used to manage power on servers plugged into one or more outlets on an IPDU appliance. The power management utility may be invoked by one of the following commands:

- pmMenu - You are presented with a menu driven interface to select the desired command.
- pmCommand - You may enter commands at the pmCommand prompt using the appropriate command arguments.

IPDU identification

When configuring and assigning names to IPDU appliances, it is important to consider the following information:

- An IPDU appliance should have a unique name, referred to as an IPDU ID.
- If the IPDU ID is not defined or is duplicated, the console server assigns a default name to an IPDU appliance.
- Once the IPDU ID is saved, the console server identifies the IPDU appliance regardless of the serial port it is connected to, or its position in the cluster.

To rename or assign a name to an IPDU:

1. From the command prompt, execute the pmCommand.

   # **pmCommand**

   Type **help** to see a list of commands or **menu** to invoke the menu driven interface.

2. To view a list of connected IPDU appliances, enter the following command.

   pmCommand> **listipdus**

3. To change an IPDU ID, enter the following command.

   pmCommand> **id <current IPDU ID> <new IPDU ID>**

NOTE: Issuing a pmCommand without an IPDU ID may apply the changes to all IPDU appliances connected to the console server, or it may generate an error message. Make sure to add the IPDU ID to a pmCommand.

IPDU appliances may also be referenced by the location of the IPDU. In this case, the location should be preceded with an exclamation (!). For example, to display the maximum detected current on the third IPDU connected to serial port 2, enter the following command at the pmCommand prompt.

pmCommand> **current !ttyS2-C**

Outlet groups should be referenced by a name which is unique among the outlet groups. For example,

pmCommand> **lock $group1**

To configure outlet groups:

1. Invoke the CLI utility and navigate to the pmdconfig directory.

   cli> **config applications pmdconfig**

2. Enter one of the following strings to add or edit outlet groups.

   pmdconfig> **outletgroups add <groupname>**

   pmdconfig> **outletgroups edit <groupname>**

3. At the Group prompt, enter the following string to assign outlets to the group.

   Group**[groupname]> outletlist <IPDU ID> [outlet name]**

   For example,

   Group**[groupname]> outletlist IPDUA[1,5],IPDU2[5-7]**

4. Activate and save your configuration.

   ```
   cli> config runconfig
   cli> config savetoflash
   ```

To configure user access to outlets:

1. Invoke the CLI utility and navigate to the pmdconfig directory.

   ```
   cli> config applications pmdconfig
   ```

2. Enter one of the following strings to add or edit outlet groups.

   ```
   pmdconfig> usermanagement add <username>
   pmdconfig> outletgroups edit <groupname>
   ```

3. Assigning outlets to the user.

   ```
   User[username]> outletlist <IPDU ID> [outlet name]
   ```

4. Activate and save your configuration.

   ```
   cli> config runconfig
   cli> config savetoflash
   ```

pmMenu

To manage IPDU appliances through pmMenu:

1. Enter **pmMenu** at the shell prompt to open the power management menu. Table 6.1 provides explanation for each menu item.
2. Select an option from the menu.
3. Follow the command instructions for the selected option.

Table 6.1: pmMenu and pmCommand Commands

Menu Item	Command Syntax	Description
Exit	exit	Exit pmMenu and return to the command prompt.
Help	help	Display a list of available commands with a description.
Who Am I	whoami	Display the name of the current user.
List IPDUs	listipdus	List the IPDUs connected to the appliance.
List Groups	listgroups	List all outlet groups.
On	on <outlet list>	Turn an outlet On. Prompts you to enter a list of <IPDU ID>[<outlet number>].
Off	off <outlet list>	Turn an outlet Off. Prompts you to enter a list of <IPDU ID>[<outlet number>].

Table 6.1: pmMenu and pmCommand Commands (Continued)

Menu Item	Command Syntax	Description
Cycle	cycle <outlet list>	Turn an outlet Off and On again (recycle power). Prompts you to enter a list of outlet numbers.
Lock (Avocent PM PDU and Cyclades IPDU)	lock <outlet list>	Lock a set of outlets in On or Off state to avoid accidental changes. Prompts you to enter a list of <IPDU ID>[<outlet number>].
Unlock (Avocent PM PDU and Cyclades IPDU)	unlock <outlet list>	Unlock the selected outlets. Prompts you to enter a list of <IPDU ID>[<outlet number>].
Status	status <outlet list>	Display the status of the selected outlets. Prompts you to enter a list of outlet numbers.
N/A	interval <outlet list> [<delay>]	Display or set the interval for an outlet to turn on.
Power On Delay (Avocent PM PDU, Cyclades IPDU and ServerTech)	powerondelay <outlet list> [<delay>]	Display or configure post turn on outlet delay. Prompts you to enter a list of outlet numbers.
Name	name <outlet entry> <outlet name>	Define a name or an alias for an outlet.
Current	current [reset] [<IPDU ID> [<element>]]	Display the amount of current that is running through the IPDU, or reset the maximum detected current in a single or all IPDU appliances. The <element> can be bank name (A, B, C, XY, ...), phase name (X, Y, Z), outlet number (1, 2, ..)
Temperature	temperature {reset} [<IPDU ID>]	Display temperature on an IPDU, if the IPDU unit is equipped with a temperature sensor. Reset the maximum detected temperature in a single or all IPDU appliances.
Humidity (Avocent PM PDU and ServerTech)	humidity {reset} [<IPDU ID>]	Display humidity or reset the maximum detected humidity in a singe or all IPDU appliances.
Voltage	voltage [<IPDU ID>]	Display voltage in a single or all IPDU appliances.
Buzzer (Avocent PM PDU and Cyclades IPDU)	buzzer {status|on|off} [<IPDU ID>]	Configure a buzzer to sound when a specified alarm threshold has reached for a single or all IPDU appliances. Options are Status, On to activate or Off to deactivate.

Table 6.1: pmMenu and pmCommand Commands (Continued)

Menu Item	Command Syntax	Description
Current Protection (Avocent PM PDU and Cyclades IPDU)	currentprotection {status|on|off} [<IPDU ID>]	Enable or disable current protection. This option is to prevent the outlets from being turned on, if the current on the IPDU exceeds the specified threshold.
Syslog (Avocent PM PDU and Cyclades IPDU)	syslog {status|on|off} [<IPDU ID>]	Enable or disable syslogging and alarm notification.
Version	ver [<IPDU ID>]	Display the software and hardware version of a single or all IPDU appliances.
Factory Default	factorydefaults [<IPDU ID>]	Restore configuration to factory default for a single or all IPDU appliances.
Reboot	reboot [<IPDU ID>]	Restart the IPDU applicances in chain.
Restore (Avocent PM PDU and Cyclades IPDU)	restore [<IPDU ID>]	Restore the configuration saved in Flash.
Save	save [<IPDU ID>]	Save the current configuration in Flash.
Alarm Threshold (Avocent PM PDU and Cyclades IPDU)	alarm <IPDU ID> <threshold> [<element>]	Set an alarm notification when the current exceeds the selected threshold. The <element> can be bank name (A, B, C, XY, ...), phase name (X, Y, Z), outlet number (1, 2, ..)
IPDU ID	id <current IPDU ID> <new IPDU ID>	Display the current IPDU name or assign a new name.
Display (Cyclades IPDU)	display [<IPDU ID> {0|180} [/{I|V}<cycle time>]]	Set the LED display mode of the IPDU.
HW OCP (Avocent PM PDU and Cyclades IPDU)	hwocp [<IPDU ID> [reset]]	Display or reset the overcurrent protection status in a single or all IPDU appliances.
Minimum On Time (Avocent SPC)	minimumon <outlet list> [<interval>]	Set the minimum time an outlet stays On before it is turned Off.
Minimum Off Time (Avocent SPC)	minimumoff <outlet list> [<interval>]	Set the minimum time an outlet stays turned Off before it is turned back On.

Table 6.1: pmMenu and pmCommand Commands (Continued)

Menu Item	Command Syntax	Description
Wake Up State (SPC and ServerTech)	wakeup <outlet list> [on | off | last]	Set the outlet state after a cold boot. It can be set to On, Off or the last saved state. [last] is for ServerTech only.
Sequence Interval	seqinterval <outlet list> [<interval>]	Set the delay time (in seconds) when turning on multiple outlets at the same time (Valid only on a master Server Technology Sentry CDU).
Cycle Interval (ServerTech)	cycleinterval <outlet list> [<interval>]	Set the time delay (in seconds) for turning on subsequent outlets after an outlet has been turned on.
(N/A)	menu	Enter menu mode (from pmCommand only).
Cold Start Delay (Avocent PM PDU)	coldstartdelay <IPDU ID> <duration>	Set the duration of the cold start delay for the defined PDU or all connected PDUs when one is not defined.
Current Threshold	currentthreshold <IPDU ID> [<element>] <thresholds>]	Display or set the threshold for current of one element Set command requires <element> argument. The <element> can be pdu, bank name (A, B, C, XY, ...), phase name (X, Y, Z), outlet number (1, 2, ...)
Power Off Delay (Avocent PM PDU)	poweroffdelay <outlet list> <delay>	Set the time delay (in seconds) for turning off outlets.
Power Info	powerinfo [reset] [<IPDU ID>] [<element>]]	Display or clear the maximum/minimum/average power consumption of a defined element or all elements if one is not defined. The <element> can be bank name (A, B, C, XY, ...), phase name (X, Y, Z), outlet number (1, 2, ..)
Cumulative Power	cumulativepower [reset] [<IPDU ID> [<element>]]	Display or clear the cumulative power consumption of a defined element, or all elements if one is not defined, to zero. The <element> can be bank name (A, B, C, XY, ...), phase name (X, Y, Z), outlet number (1, 2, ..)
Power Factor	powerfactor [reset] [<IPDU ID> [<element>]]	Display or reset the maximum/minimum/average recorded power factor of a defined element or all elements if one is not defined. The <element> can be bank name (A, B, C, XY, ...), phase name (X, Y, Z), outlet number (1, 2, ...)

Table 6.1: pmMenu and pmCommand Commands (Continued)

Menu Item	Command Syntax	Description
Voltage Info	voltageinfo [reset] [<IPDU ID> [<element>]]	Display or clear the maximum/minimum/average recorded voltage for the defined element, or all elements if one is not defined, to zero. The <element> can be bank name (A, B, C, XY, ...), phase name (X, Y, Z), outlet number (1, 2, ...)
Sensors	sensors [reset] [<IPDU ID> [<sensor name>]]	Display or clear the maximum/minimum/average recorded <type> for the defined sensor to zero.
Sensors Unit	sensors unit [<IPDU ID>] <unit>	Display or set the unit (Celsius or Fahrenheit) for the temperature sensor.
Sensors Threshold	sensors threshold <IPDU ID> <sensor name> <high critical> <high warn> <low warn> <low critical>	Display or set the environment monitoring thresholds. For set, all arguments are required.

pmCommand

Alternatively, you can use the pmCommand to manage IPDU appliances. Refer to Table 6.1 for command syntax and arguments.

```
Usage: pmCommand [<command> [<arguments>]]
```

To manage IPDU appliances through pmCommand:

1. Enter the following command at the shell prompt to invoke the power management command utility.

   #**pmCommand**

2. At the pmCommand prompt, type **help** to see a list of commands along with a description, or type **menu** to invoke the menu driven interface.
3. Alternatively, if you know the specific command and argument, enter it with the following format.

   # **pmCommand[<command>[<arguments>]]**

To manage power through the console server:

1. From the console server, open a Telnet or SSH session to the serial port where your server is connected.
2. Access the IPDU by entering the preconfigured hotkey. The default is **^p**.
   a. If you have permission to access the server outlet(s), the IPDU appliance menu appears. Table 6.2 describe the commands available through the menu.
   b. If you do not have permission to access an outlet, the following message appears.

```
It was impossible to start a Power Management Session
You can't access any Power Management functionality.
Please contact your Console Server Administrator.
```

c. If you can access outlet(s) but have no access to outlet(s) of a specific server, the following message appears.

```
You cannot manage the outlet(s) of this server.
Please enter the outlet(s) (or '?' for help):
```

3. Enter the outlet(s) you want to manage. The main menu appears only if you have permission for those outlet(s). Type **h** to display the help information.

Table 6.2: IPDU Appliance Command Menu

Command	Description
Exit	Exits the power management session.
Help	Display a list of available commands with a description.
On	Turn outlet(s) On.
Off	Turn outlet(s) Off.
Cycle	Turn an outlet Off and On again (recycle power).
Lock	Locks a set of outlets in On or Off state to avoid accidental changes.
Unlock	Unlocks the selected outlets.
Status	Displays the status of the selected outlets.
Interval	Configures the post turn on delay.
Other	Allows you to manage other outlets.

4. Check the status of the server's outlet(s) by typing **8** to select Status**.**
5. If the outlet(s) are locked you must unlock them first. Type **7** to select Unlock.
6. The Cycle command turns off the outlet for a few seconds and turns it back on. Type **5** to select Cycle.

To manage other outlets:

Perform the following procedures if you need to access other outlets.

1. Type **8** to select Status to view the outlets you are authorized to manage.
2. Type **10** to select Other, and select the outlet you want to manage. You should have authorization to the manage the outlets entered.

IPDU password

Although you may not be required to change an IPDU password, you can perform the following procedure to change a password, if needed.

To change an IPDU password:

1. Change the connection protocol of the serial port where the IPDU appliance is connected.

   cli> **config physicalports <port number> general protocol consoletelnet**

2. Activate your configuration changes.

   cli> **config runconfig**

3. Exit the CLI utility, and from the command prompt, telnet to the console server serial port where the IPDU appliance is connected.

   #**telnet localhost <TCP Port>**

4. Log in to the IPDU appliance with the current username and password. The default value for the Avocent PM PDU and Cyclades IPDU is **admin/pm8**.
5. At the pm prompt change the IPDU appliance password and save the new password.

   pm> **passwd <new password>**

   pm> **save**

6. Press **Ctrl-]** (**^]**) to quit the Telnet session.
7. Invoke the CLI utility and change the connection protocol to power management.

   cli> **config physicalports <port number> general protocol pm**

8. Repeat steps 1-7 for each Cyclades IPDU appliance.
9. Configure and synchronize the new password in the IPDU appliance with the password stored in the console server.

   cli> **config applications pmdconfig general cyclades password <new password>**

10. Activate your configuration changes.

    cli> **config runconfig**

11. Execute the following command to check if all Cyclades IPDU appliances are detected.

    #**pmCommand listipdus**

IPDU Firmware Upgrade

You may upgrade the firmware of a single or multiple daisy-chained IPDU power management appliances connected to a serial port of the console server.

NOTE: The firmware upgrade is available for Avocent PM PDUs and for Cyclades PM IPDUs.

To upgrade IPDU firmware:

1. From http://www.avocent.com, click *Resources-Updates and Documentation.*
2. From the Updates section, click *Cyclades PM Intelligent Power Distribution Unit for Cyclades PM IPDU* or click *PM2000 PM3000 Rack PDUs for Avocent PM PDU*. The web page will show a firmware list.
3. Click the *firmware* link. It is recommended that you download the new firmware to a /tmp directory since files in this directory are deleted during the boot process.

NOTE: It is recommended that you run md5sum on the file after you download it and compare the md5sum output with the contents of the firmware md5 file on the avocent web site, to ensure that the firmware file you downloaded was not corrupted.

CAUTION: It is possible that all outlets get turned off during the upgrade process. Make sure to shut down all connected devices before starting the firmware upgrade process.

4. Execute the pmfwupgrade application from the shell prompt. Table 6.3 describe the parameters.

   # **pmfwupgrade [<options>] {all | -s <serial device name> | <IPDU id>} [<filename>]**

Table 6.3: pmfwupgrade Application Parameters

Parameter	Description
-h	Show the help message and exit.
-f	Upgrade the firmware without asking any questions.
-v	Show messages about the status of the upgrade.
-s	<serial device name>: Serial port name where the PM IPDU is connected. This option upgrades all IPDU appliances in daisy-chained IPDUs connected to the serial port.
<IPDU id>	IPDU identification name.
<filename>	The new firmware to upload to the PM unit. Default:/tmp/pmfirmware.

SNMP Proxy

The SNMP proxy for power management feature allows the console server to proxy SNMP requests to the IPDU. This allows SNMP clients to query and control the remote IPDU using standard set and get commands.

The following parameters and features are controlled in the remote IPDU:

- The number of IPDU units connected to the console server.
- The number of outlets connected to a serial port.

- The number of IPDU units connected to a serial port in a daisy-chain configuration.
- The instantaneous RMS current being drawn from each of the IPDU connected to a serial port.
- The software version of the IPDU connected to a serial port.
- The information about sensors (Current, Voltage, Power Factor) and the Power Consumption of the PDU and for each element (banks, phases, outlets).
- The name of the outlet as configured in the IPDU.
- The alias of the server that is configured for using the IPDU outlet.
- The name of the console server to which the IPDU is connected.
- The status of the outlet:
  - Power status : 0 (Off), 1 (On), 3 (unknown)
  - Lock state : 0 (Unlock), 1 (Lock), 2 (unknown)

SNMP proxy allows an administrator to control the IPDU outlets using SNMP set commands. The SNMP commands that may be executed on each outlet are ON, OFF, CYCLE and LOCK.

NOTE: The console server proxies all SNMP requests to the IPDU. Therefore, there is a small delay if an outlet-cycling is requested by the snmpset command. To successfully cycle an outlet, a four second or higher time-out must be specified. To run this command for more than one outlet or for units configured as daisy-chain, this time should be recalculated.

To configure SNMP proxy:

The following example shows how to configure this feature.

1. Get the console server serial port number to which the IPDU is connected.

   # **snmpget -m all -v 2c -t 4 -c cyclades 10.10.0.1 .cyNumberOfPM**

   Enter

   ```
   CYCLADES-ACS5K-PM-MIB::cyNumberOfPM.0 = 2
   ```

2. Get the number IPDU outlets connected to the serial port 16.

   # **snmpget -m all -v 2c -t 4 -c cyclades 10.10.0.1 .cyPMNumberOutlets.16**

   Enter

   ```
   CYCLADES-ACS5K-PM-MIB::.cyPMNumberOutlets.16 = 8
   ```

3. Get the number of IPDU units connected to serial port 14.

   # **snmpget -m all -v 2c -t 4 -c cyclades 10.10.0.1 .cyPMNumberUnits.14**

   Enter

   ```
   CYCLADES-ACS5K-PM-MIB::.cyPMNumberUnits.14 = 2
   ```

APPENDICES

Appendix A: Additional Features and Applications

Windows 2003 server management

Emergency Management Services (EMS) is a new feature in the Windows 2003 Server that allows out-of-band remote management and system recovery tasks. All Emergency Management Services output is accessible using a terminal emulator connected to the server serial port. Besides the normal character mode output sent to the serial console, Windows also sends xml tags. Those tags may be captured and processed by the console server so that the administrator may automate the actions to be taken.

You may manage the server through the Special Administration Console (SAC), which is the console when connected directly to the Windows Server through Telnet or SSH session.

Configuring Windows 2003 server management

To manage a Windows 2003 server it is necessary to enable the EMS service.

Syntax

```
bootcfg /ems [EDIT|OFF|ON] [/s [server] [/u [[domain\]user] /p
password [/baud baud_rate] [/port communications_port] /id line_number
```

Table A.1: EMS Configuration Parameters and Switches

Parameter and Switches	Description
EDIT	Allows changes to port and baud rate settings by changing the redirect=COMx setting in the [bootloader] section. The value of COMx is set to the value of the /port.
OFF	Disables output to a remote server. Removes the /redirect switch from the specified line_number and the redirect=comX setting from the [boot loader] section.
ON	Enables remote output for the specified line_number. Adds a /redirect switch to the specified line_number and a redirect=comX setting to the [boot loader] section. The value of comX is set by the /port.
/ems	Enables the user to add or change the settings for redirection of the EMS console to a remote server. By enabling EMS, you add a redirect=Port# line to the [boot loader] section of the BOOT.INI file and a /redirect switch to the specified operating system entry line. The EMS feature is enabled only on servers.

Table A.1: EMS Configuration Parameters and Switches (Continued)

Parameter and Switches	Description
/baud_rate	Specifies the baud rate to be used for redirection. Do not use if remotely administered output is being disabled. Valid values are: 9600, 19200, 38400, 57600, 115200.
/id line_number	Specifies the operating system entry line number in the [operating systems] section of the Boot.ini file to which the operating system load options are added. The first line after the [operating systems] section header is 1.
/p password	Specifies the password of the user account that is specified in /u.
/port communications_port	Specifies the COM port to be used for redirection. Do not use if remotely administered output is being disabled. BIOSSET get BIOS settings to determine port COM1 COM2 COM3 COM4
/s server	Specifies the name or IP address of a remote server (do not use backslashes). The default is the local server.
/u [[domain\]user]	Runs the command with the account permissions of the user specified by User or Domain\User. The default is the permissions of the current logged on user on the server issuing the command.

With the EMS service enabled in Windows, configure the console server as console profile to manage the Windows 2003 server. Windows sends xml tags in the following situations.

- During Windows installation, it sends <channel-switch> with the setup logs.
- During boot, it sends the <machine-info> information.
- When switching channels, it sends the <channel-switch> information.
- During system crash, it sends the <BP> to indicate BreakPoint.

The <machine-info> tag is emitted once by Windows Server during its system boot sequence. This tag is also emitted as part of the <BP> tag. The following elements are included in <machine-info> tag.

Table A.2: Machine Info Tags

Element	Description
<guid>	It is the GUID that uniquely identifies the server platform. Normally, this is an SMBIOS provided identification. If no such value is available, all 0's GUID string is used. See *Example of sample encoding* on page 113.
<name>	Is the system name.

Table A.2: Machine Info Tags (Continued)

Element	Description
<os-build-number>	Is a numeric string that identifies a successive Windows Build.
<os-product>	Is the name of the Windows Server 2003 product currently running on this server. It is one of the following. • Windows Server 2003 Datacenter Edition • Windows Server 2003 Embedded • Windows Server 2003 Enterprise Edition • Windows Server 2003
<os-service-pack>	Is an alphanumeric string that identifies the most up-to-date service pack installed. If none installed, the string is None.
<os-version>	Is the numeric identification of the Windows version currently running.
<processor-architecture>	Is either x86 or IA64, designating the two processor architectures currently supported by Windows Server 2003.

Example of sample encoding

```
<?xml>
<machine-info>
<name>NTHEAD-800I-1</name>
<guid>00000000-0000-0000-0000-000000000000</guid>
<processor-architecture>x86</processor-architecture>
<os-version>5.2</os-version>
<os-build-number>3735</os-build-number>
<os-product>Windows Server 2003 Enterprise Edition</os-
product>
<os-service-pack>None</os-service-pack>
</machine-info>
```

In the SAC command line, each time you enter the cmd command you create a channel. A channel is the Command Prompt environment where you may enter the command prompt commands such as dir, cd, edit, del or copy. You may switch back and forth between channel(s) and SAC by pressing **Esc** or **Tab** keys. You may create up to nine channels (nine command prompt sessions.) Whenever we switch channels, the <channel-switch> tag is sent.

The following elements are included in the <channel-switch> tag.

Table A.3: Elements in the <channel-switch> Tag

Element	Description
<application-type>	Is a hexadecimal GUID signifying the application or tool that is running on the Windows Server platform and communicating via this active channel. It is to be used to discern the different interaction modes. During the Windows GUI-mode Setup phase, the following GUIDs identify the specific types of data being emitted. • Debug Log (5ED3BAC7-A2F9-4E45-9875-B259EA3F291F) • Error Log (773D2759-19B8-4D6E-8045-26BF38402252) • Action Log (D37C67BA-89E7-44BA-AE5A-112C6806B0DD) During nominal Windows Server operations, the following GUIDs may be expected. • SAC (63D02270-8AA4-11D5-BCCF-806D6172696F) • CMD (63D02271-8AA4-11D5-BCCF-00B0D014A2D0) **NOTE:** These GUIDs are constant and should not be confused with those provided through the <guid> tag.
<description>	Is the user-friendly name of the active channel. For the GUI-Mode Setup tool they are, • Debug Log (Setup tracing log) • Error Log (Setup errors log) • Action Log (Setup actions log) For the Windows Server, they are, • SAC (Special Administration Console) • CMD (Command Prompt)
<guid>	Is a hexadecimal GUID that identifies a specific instance of a channel. During a life-span of a Windows Server (between any two system boots), there is a total of 10 channels being allocated, one of those may be a GUID for each of the following channel types. • GUI-Mode Setup Debug Log • GUI-Mode Setup Error Log • GUI-Mode Setup Action Log • SAC The remaining GUIDs are of the CMD channel type. For example, during Windows setup, there are 3 GUIDs assigned to Setup, 1 to SAC and the remaining 6 to CMD. However, during normal Windows operations, there is 1 GUID assigned to SAC and the remaining 9 to CMD. These GUIDs are created a new for each instance of channels, and should not be confused with the constant GUIDs provided via the <application-type> tag listed previously.

Table A.3: Elements in the <channel-switch> Tag (Continued)

Element	Description
<name>	Is the system name of the active channel. For the GUI-mode Setup tool, they are the filenames where the data is written. • Debug Log (setuplog.txt) • Error Log (setuperr.log) • Action Log (setupact.log) For Windows Server, they are, • SAC (SAC) • CMD (Cmdnnnn), where nnnn indicates the corresponding channel number
<type>	Is the type of data being emitted on the active channel. Currently, there are two types of data supported. • Raw for the 3 GUI-Mode Setup channels • VT-UTF8 for the SAC and CMD channels

A sample encoding of the SAC channel tag follows.

```
<channel-switch>
<name>SAC</name>
<description>Special Administration Console</description>
<type>VT-UTF8</type>
<guid>1aee4cc0-cff3-11d6-9a3d-806e6f6e6963</guid>
<application-type>63d02270-8aa4-11d5-bccf-806d6172696f</application-type>
</channel-switch>
```

A sample encoding of the CMD channel tag follows.

```
<channel-switch>
<name>Cmd0001</name>
<description>Command Prompt</description>
<type>VT-UTF8</type>
<guid>970438d1-12bb-11d7-8a92-505054503030</guid>
<application-type>63d02271-8aa4-11d5-bccf-00b0d014a2d0</application-type>
</channel-switch>
```

A sample encoding of the GUI mode Setup Debug Log channel tag follows.

```
<channel-switch>
<name>setuplog.txt</name>
<description>Setup tracing log</description>
<type>Raw</type>
<guid>6f28e904-1298-11d7-b54e-806e6f6e6963</guid>
<application-type>5ed3bac7-a2f9-4e45-9875-b259ea3f291f</application-type>
</channel-switch>
```

The <BP> tag is emitted when the Windows server system halts such that only elements of the kernel are the most recently operating logic.

Table A.4: <BP> Tags Description

Element	Description
<INSTANCE CLASSNAME=>	Is the type of break point. Currently, there is only one type emitted such as Blue Screen which indicates the system was halted prematurely. It is represented by the CLASSNAME=”BLUESCREEN” value.
<machine-info>	Described previously.
<PROPERTY NAME=>	Provides additional details, such as error code of the abnormal condition that caused the break point.

A sample encoding of the Break Point tag follows.

```
<?xml>
<BP>
<INSTANCE CLASSNAME="BLUESCREEN">
<PROPERTY NAME="STOPCODE"
TYPE="string"><VALUE>"0xE2"</VALUE>
</PROPERTY>
<machine-info>
<name>NTHEAD-800I-1</name>
<guid>00000000-0000-0000-0000-000000000000</guid>
<processor-architecture>x86</processor-architecture>
<os-version>5.2</os-version>
<os-build-number>3735</os-build-number>
<os-product>Windows Server 2003 Enterprise Edition</os-
product>
<os-service-pack>None</os-service-pack>
</machine-info>
</INSTANCE>
</BP>
```

XML monitoring parameters in pslave.conf

Some parameters need to be configured in the /etc/portslave/pslave.conf to monitor XML data. For instance, for ttyS1 configure the following parameter.

```
s1.xml_monitor        1
```

When the xml_monitor is set, cy_buffering searches for xml packets from the serial port. When a complete xml packet is received, cy_buffering sends it to syslog-ng. In syslog-ng.conf, the following filters are available to filter the xml messages.

```
filter f_windows_bluescreen { facility(local<conf.DB_facility>) and
level(info)nd match("XML_MONITOR") and match("BLUESCREEN"); } ;
```

- and -

```
filter f_windows_boot { facility(local<conf.DB_facility>) and
level(info) and match("XML_MONITOR") and
not match("BLUESCREEN") and match("machine-info"); } ;
```

Once the desired message is filtered, define which actions we would like to take. Syslog-ng creates macros that may give easy access for the administrators to access the xml information. If the administrator uses these macros, syslog-ng replaces the macros by the data received in the xml packet.

Table A.5 shows the macros that are available when filter f_windows_bluescreen is successful, and the examples of values that may replace the macros.

Table A.5: f_windows_boot Macros

Macro	Description	Value to replace macro
$<INSTANCE CLASSNAME=>	Reason for the break point. Currently there is only one type, BLUESCREEN.	BLUESCREEN
$<PROPERTY NAME=>	Additional details about break point.	STOPCODE
$<VALUE>	Additional details about break point.	0xE2
$<name>	Machine name.	MY_WIN_SERVER
$<guid>	GUID that uniquely identifies this server. If no such value is available, all 0's GUID string is used.	4c4c4544-8e00-4410-8045-80c04f4c4c20
$<processor-architecture>	Processor architecture. It may be either x86 or IA64.	x86
$<os-version>	Windows version.	5.2
$<os-product>	Which Windows Server product. It may be Windows Server 2003 Datacenter Edition, Windows Server 2003 Embedded, Windows Server 2003 Enterprise Edition or Windows Server 2003.	Windows Server 2003
$<os-service-pack>	Alphanumeric string that identifies the most up-to-date service pack installed. If none installed, the string is None.	None
$<tty>	console server serial port tty or alias name.	S1.ttyS1

For the f_windows_boot, the following macros are available.

Table A.6: f_windows_boot Available Macros

Macro	Description	Value to replace macro
$<name>	Machine name	MY_WIN_SERVER
$<guid>	GUID that uniquely identifies this server. If no such value is available, all 0's GUID string is used.	4c4c4544-8e00-4410-8045-80c04f4c4c20
$<processor-architecture>	Processor architecture. It may be either x86 or IA64.	x86
$<os-version>	Windows version.	5.2
$<os-build-number>	Numeric string that identifies a successive Windows Build.	3763
$<os-product>	Which Windows Server product. It may be Windows Server 2003 Datacenter Edition, Windows Server 2003 Embedded, Windows Server 2003 Enterprise Edition or Windows Server 2003.	Windows Server 2003
$<os-service-pack>	Alphanumeric string that identifies the most up-to-date service pack installed. If none installed, the string is None.	None
$<tty>	console server serial port tty or alias name.	S2.server_connected_to_serial2

An example on how to use the macros

In the following example, the console server sends an email to the administrator whenever a crash happens. The email should have the information about the reason of the crash, machine name and windows version information. The following entry should be created in syslog-ng.conf.

```
destination win2003mail { pipe("/dev/cyc_alarm"
template("sendmail -t administrator@cyclades.com -f acs -s
\"\
Server $<name> crashed\" -m \'\
Break Point: $<INSTANCE CLASSNAME=>  $<PROPERTY NAME=>
$<VALUE>\
Server: $<name>\
OS: $<os-product>\
Build: $<os-build-number>  Version: $<os-version>\
Service Pack: $<os-service-pack>\
Processor: $<processor-architecture>\
Server GUID: $<guid>\
ACS port: $<tty>\
\' -h mail.cyclades.com "));};
```

The following entry activates the win2003mail action when the f_windows_bluescreen filter is successful.

```
source src { unix-stream("/dev/log"); };

log { source(src); filter(f_windows_bluescreen);
destination(win2003mail); };
```

Server commands

The following are the commands that may be sent to the server.

Table A.7: Server Commands

Command Set	Description
ch	Channel management commands.
ch -ci <#>	Close a channel by its number.
cmd	Create a Command Prompt channel.
ch -si <#>	Switch to another channel (from Channel 0).
d	Dump the current kernel log.
f	Toggles the information output by the t-list command, which shows processes only, or shows processes and threads.
i	List all IP network numbers and their IP addresses.

Table A.7: Server Commands (Continued)

Command Set	Description
i <#> <ip> <subnet> <gateway>	Set network interface number, IP address, subnet and gateway.
id	Display the server identification information.
k <pid>	Kill the given process.
l <pid>	Lower the priority of a process to the lowest possible.
lock	Lock access to Command Prompt channels. You must provide valid logon credentials to unlock a channel.
m <pid> <MB-allow>	Limit the memory usage of a process to <MB-allow>.
p	Causes t-list command output to pause after displaying one full screen of information.
r <pid>	Raise the priority of a process by one.
s	Display the current time and date (24 hour clock used).
mm/dd/yyyy hh:mm	Set the current time and date (24 hour clock used).
t	Tlist.
crashdump	Crash the system. Crash dump must be enabled.
restart	Restart the system immediately.
shutdown	Shut down the system immediately.

Intelligent Platform Management Interface (IPMI)

IPMI is a service-level protocol and implementation that provides intelligent management to servers. IPMI allows server control and monitoring by means of an always-on chip located on the server's motherboard called the Baseboard Management Controller (BMC) that may respond to IPMI commands out-of-band.

The Cyclades ACS 5000 advanced console server has an implementation of IPMI over LAN, which allows the console server to control power on servers, and also to obtain sensor readings such as CPU temperature or fan speed.

The IPMI support in the console server, extends its functionality so the console server may be used to control power to the serially connected servers through the IPMI protocol.

IPMI configuration

This program lets you manage IPMI enabled devices locally remotely. These functions include printing FRU information, LAN configuration, sensor readings and remote chassis power control.

IPMI [ipmitool]

Syntax

```
ipmitool [-hvV] -I interface -H hostname [-L privlvl] [-A authType] [-
P password] <expression>
```

Table A.8: ipmitool Options

Option	Description	Valid Values
-h	Get basic usage help from the command line.	N/A
-v	Increase verbose output level. This option may be specified multiple times to increase the level of debug output.	N/A
-V	Display version information.	N/A
-I <interface>	Selects IPMI interface to use.	open | imb | lan | lanplus
-H <address>	Remote server address, may be IP address or hostname. This option is required for the LAN interface connection.	N/A
-U <username>	Remote username.	Default is NULL.
-L <privlvl>	Force session privilege level.	USER | OPERATOR | ADMIN. Default is USER
-A <authtype>	Force session authentication type.	PASSWORD | MD5 | MD2
-P <password>	Remote server password.	Valid password for specified username account.

Expressions

Table A.9: IPMI Commands

Expression	Description
raw	Send a RAW IPMI request and print response
i2c	Send an I2C Master Write-Read command and print response
lan	Configure LAN Channels
chassis	Get chassis status and set power state
event	Send pre-defined events to MC
mc	Management Controller status and global enables
sdr	Print Sensor Data Repository entries and readings
sensor	Print detailed sensor information
fru	Print built-in FRU and scan SDR for FRU locators
sel	Print System Event Log (SEL)
pef	Configure Platform Event Filtering (PEF)
sol	Configure IPMIv2.0 Serial-over-LAN
isol	Configure IPMIv1.5 Serial-over-LAN
user	Configure Management Controller users
channel	Configure Management Controller channels
session	Print session information
sunoem	OEM Commands for Sun servers
exec	Run list of commands from file
set	Set runtime variable for shell and exec

To Configure IPMI:

1. Enter the following command to enable IPMI or edit an existing configuration.

   cli> **config ipmi [add |edit|delete]**

2. Configure serial ports for power management and IPMI protocol. Refer to Table A.10 for configuration parameters.

   cli> **config physicalports <'all' or range/list[1-xx] powermanagement [disableIPMI | enableIPMI] <key <^(character)>> <server <name>>**

Table A.10: IPMI CLI Configuration Parameters

Parameter	Value	Description
add	<alias>	Add and configure an IPMI device.
edit	<alias>	Edit the parameters of an IPMI enabled device.
delete	<alias>	Remove an IPMI device.
authtype	md2 md5 none password	Add an authentication method.
password	<password>	Assign a password to access the device.
privilege	admin operator user	Assign a user access level.
serverip	<n.n.n.n>	IP address of the device.
username	<name>	Username to access the device.
key	<^(character)>	The hotkey used to access the IPMI. **NOTE:** The default IPMI hotkey is **^I**, where **^** stands for the **Ctrl** key on the keyboard. The hexadecimal code for the **Ctrl+I** default IPMI hotkey is the same as the keyboard's **Tab** key. You may choose to change the default using this parameter.
server	<alias>	The name of the IPMI device.

3. Activate and save your configuration.

Line printer daemon

This feature implements the UNIX Line Printer Daemon (LPD) in the console server and may be used with local serial printers. It enables the console server to receive network print requests and service them using locally attached serial printers.

To configure the lpd:

1. Setup the serial port where the serial printer is connected.
2. Edit the /etc/portslave/pslave.conf file and set the protocol of the serial port to lpd.

   ```
   s2.protocol    lpd
   ```

3. Create the printer definition. Edit the /etc/printcap file and configure the printer. The spool directory is created automatically by cy_ras process.

Example

```
#comment
# primary printer name and alias
# lp |lp2| serial printer on port ttyS2
#suppress header and/or banner page
#:sh:
#spool directory - the name is fixed as lp_ttySnn when nn is
the
#serial port number
#:sd=/var/spool/lpd/lp_ttyS2:
#printer device
#:lp=/dev/ttyS2:
#log filename
#:lf=/var/log/lpd.log:
#set serial port speed as 115.200 bps
#:br115200:
lp|lp2| serial printer on port ttyS2:\
:sh: \
:sd=/var/spool/lpd/lp_ttyS2: \
:lp=/dev/ttyS2: \
:lf=/var/log/lpd.log:
```

4. Enable the printer daemon file /etc/lpd.sh and change the option ENABLE to YES.
5. Allow clients to use the service.

   Edit the file /etc/hosts.lpd and include the host names that have permission to use the console server printers.

NOTE: The lpd needs to translate the IP address of the request message to the host name, check your resolv.conf file.

6. Restart the processes by executing the commands runconf and daemon.sh.

7. Execute saveconf command to save the configuration in Flash.
8. Check the console server configuration by entering the following command at your workstation.

   # **lpr -P lp@<ACS IP address> <file that you want printer>**

CAS port pool

CAS port pooling allows you to access a free serial port from a pool in addition to the original feature where you could access a specific serial port. When you access a serial port through the pool sniff session and multiple sessions, features are not available. This feature is available for serial ports configured as CAS profile only.

You may define more than one pool of serial ports. Each serial port may only belong to one pool. The pool is uniquely identified by a four parameter scheme.

- protocol
- pool_ipno
- pool_alias
- pool_socket_port

The three parameters pool_ipno, pool_alias and pool_socket_port have the same meaning as ipno, alias and socket_port respectively. Ports that belong to the same pool must be configured with the same value.

It is strongly recommended that you configure the same values in all parameters related to authentication for all serial ports belonging to a pool. You may access the serial ports from a pool with the same commands you use to access a specific serial port. You need to use pool_ipno, pool_alias or pool_socket_port instead of ipno, alias or socket_port with an SSH or Telnet command.

When a connection request arrives using one of pool_ipno, pool_alias, or pool_socket_port, the console server looks for the first free serial port from the pool and assigns it to the connection. If there is no free serial port in the pool, the connection is dropped.

To configure port pooling:

Configuration is made in the /etc/portslave/pslave.conf file. Don’t forget to activate and save the configuration by issuing the commands runconf and saveconf respectively.

The following is an example of serial port pool configuration. In this example there are two pools.

- pool-1 (identified by Protocol socket_server, TCP port #3000, IP 10.1.0.1 and alias pool-1)
- pool-2 (identified by Protocol socket_ssh, TCP port #4000, IP 10.2.0.1 and alias pool-2)

The serial ports ttyS1 and ttyS2 belong to pool-1. The serial ports ttyS3 and ttyS4 belong to pool-2.

You may access serial port ttyS1 by using TCP port 7001, IP address 10.0.0.1 or alias serial-1. If the ttyS1 is in use and if the user is not an admin user, then the connection is dropped.

Alternately, you may access ttyS1 through the pool using TCP port 3000, IP 10.1.0.1 or alias pool-1. If it is not free ttyS2 is automatically allocated. If ttyS2 is not free, then the connection is dropped.

```
# Serial port pool: pool-1
#
s1.tty ttyS1
s1.protocol socket_server
s1.socket_port 7001 // TCP port # for specific allocation
s1.pool_socket_port 3000 // TCP port # for the pool
s1.ipno 10.0.0.1 // IP address for specific allocation
s1.pool_ipno 10.1.0.1 // IP address for the pool
s1.alias serial-1 // alias for specific allocation
s1.pool_alias pool-1 // alias for the pool
s2.tty ttyS2
s2.protocol socket_server
s2.socket_port 7002 // TCP port # for specific allocation
s2.pool_socket_port 3000 // TCP port # for the pool
s2.ipno 10.0.0.2 // IP address for specific allocation
s2.pool_ipno 10.1.0.1 // IP address for the pool
s2.alias serial-2 // alias for specific allocation
s2.pool_alias pool-1 // alias for the pool
#
# Serial port pool: pool-2
#
s3.tty ttyS3
s3.protocol socket_ssh
s3.socket_port 7003 // TCP port # for specific allocation
s3.pool_socket_port 4000 // TCP port # for the pool
s3.ipno 10.0.0.3 // IP address for specific allocation
s3.pool_ipno 10.2.0.1 // IP address for the pool
s3.alias serial-3 // alias for specific allocation
s3.pool_alias pool-2 // alias for the pool
s4.tty ttyS4
s4.protocol socket_ssh
s4.socket_port 7004 // TCP port # for specific allocation
s4.pool_socket_port 4000 // TCP port # for the pool
s4.ipno 10.0.0.4 // IP address for specific allocation
s4.pool_ipno 10.2.0.1 // IP address for the pool
s4.alias serial-4 // alias for specific allocation
s4.pool_alias pool-2 // alias for the pool
```

Billing

The console server family of products may be used as an intermediate buffer to collect serial data (like billing tickets from a PBX), making them available for a posterior file transfer. Different ports may have simultaneous billing sessions.

NOTE: Billing is supported only on ACS 5000 Advanced Console Servers running firmware version 3.3.x or earlier.

General feature description

The console server reads the serial port and saves the information to Ramdisk files, which is limited to the maximum number of records per file. After the files are closed, they are available for transfer at /var/run/DB or an alternate path defined by the user in the pslave.conf file.

Once the cy_ras program detects the protocol as billing, it starts the billing application. The billing application then opens the port (as configured in pslave.conf) and starts reading it. Records terminated by billing_eor string are expected to be received. The console server doesn't change the termination method, transferring the same sequence to the file. The name of the temporary file used to write these records is,

```
cycXXXXX-YYMMDD.hhmmss.tmp
```

- where -

```
XXXXX is the "hostname" or "alias"
YYMMDD is the year/month/day
hhmmss is the hour:min:sec
```

This name helps the user archive and browse their directory as the file may be chronologically listed, not based on its creation or modification times, but based on when its contents were recorded. Also, whenever hostname is not significant, the user may use the alias name (s1.alias in pslave.conf) to match their actual plant (like PABX-trunk9). The temporary file described previously is closed and renamed to cycXXXXX-YYMMDD.hhmmss.txt and a new temporary file is opened when,

- the maximum number of records specified by billing_records is reached
- the lifetime specified by billing_timeout finishes

If no record is received within a file lifetime period, no file is saved.

NOTE: A zero-value for billing_record stops the application and a zero-value for billing_timeout means no time-out is desired. The file is closed after billing_records are received.

To configure billing:

1. Open the /etc/portslave/pslave.conf file and configure the following parameter according to your application.

   ```
   all.protocol - billing
   ```

2. In the data buffering section of pslave.conf file configure the following parameters.

   ```
   all.billing_records - 50
   all.billing_timeout - 60 min
   all.billing_eor - "\n"
   ```

NOTE: The values presented implement the billing feature for all ports of the product. If the configuration for a specific port is required, all related parameters beginning with all must be changed to S.x, where x is the number of the port to be configured.

Disk space issue

It is important to note that there is protection against disk space problems. If you configure flow control to hardware for the serial port (all.flow = hard in the pslave.conf file), the application monitors the available disk space and if it is less than 100 Kb, the serial interface deactivates RTS signal on the RS-232. RTS is reactivated once the disk free space is greater than 120 Kb.

Billing wizard

This feature improves the billing application by using a script and automating the upload of the billing records files from the console server to a remote server using FTP or SSH.

config_billing.sh script

The config_billing.sh script is used to configure a serial port for billing protocol, and configure upload scripts using FTP or SSH. The config_billing.sh script configures the files /etc/billing_up.conf /etc/billing_crontab, and /etc/crontab_files.

```
Usage: config_billing.sh [X] [options]

X is the port number to be configured
[options]
-s speed
-d data size
-b stopbit
-p parity
-r billing records
-e billing EOR (this parameter must be on " ", like "\n")
-D billing dir
-S serverFar
-t date
```

```
-T timeout
-i ip
-n netmask
-R route
-u upload
```

Any parameter that is not specified remains unchanged. The following parameters are configured by default for billing.

```
sxx.authtype none
sxx.protocol billing
sxx.flow none
sxx.dcd 0
sxx.sniff_mode no
```

Select the -u option to execute the billing_upload_files.sh script. The script presents the following sequential menu where the upload options may be configured.

```
# billing_upload_files.sh
Transfer Mode (ftp or scp)[ftp]:
Local Directory[/var/run/DB]:
Remote server IP [192.168.1.101]:
Remote directory [/var/billing]:
User [billing]:
Password [billing]:
Upload Interval in minutes []:
```

NOTE: Instead of running the -u option, the /etc/billing_up.conf may be configured manually to change the parameters. If the parameters remain unchanged, the default parameters are uploaded.

NOTE: If the scp transfer mode is selected and there is no defined authentication, the script generates a key and uploads to the server. The key must be stored on the server with the appropriate configuration.

Execute saveconf and restart the console server to activate the options related to billing upload.

Appendix B: Upgrades and Troubleshooting

Upgrades

Below are the six files added to the standard Linux files in the /mnt/flash directory when an upgrade is needed.

- boot_alt - alternate boot code
- boot_conf - active boot code
- boot_ori - original boot code
- config.tgz - console server configuration information
- zImage - Linux kernel image

To upgrade the console server:

1. Log in to the console server as root.
2. Go to /mnt/flash.
3. FTP to the host where the new firmware is located.
4. Log in to the FTP server and go to the directory where the firmware is located.

```
# ftp
ftp> open server
ftp> user admin
ftp> Password: adminpw
ftp> cd /tftpboot
ftp> bin
ftp> get zImage.nnn zImage
ftp> quit
```

NOTE: The destination filename in the /mnt/flash directory must be zImage. Example (hostname = server; directory = /tftpboot; username= admin; password = adminpw; firmware filename on that server = zImage.*nnn*).

NOTE: Due to space limitations, the new zImage file may not be downloaded with a different name, then renamed. The console server searches for a file named zImage when booting and there is no room in Flash for two zImage files.

5. To make sure the downloaded file is not corrupted and to verify the zImage saved in Flash, run the following command.

   ```
   # md5sum /mnt/flash/zImage
   ```

   The system responds with a message similar to the following.

   ```
   5bcc7d9b3c61502b5c9269cbecd20317  /mnt/flash/zImage
   ```

6. Check the system's response against the .md5 text file on the tftp server.

For example, the <zImage_filename.md5> text file contains information similar to the following.

```
5bcc7d9b3c61502b5c9269cbecd20317 /tftpboot/<zImage_filename>
```

7. If the alphanuemeric string matches the downloaded file, execute the reboot command.
8. After reboot, the console server is updated with the new firmware. Confirm by issuing the following command.

   ```
   # cat /proc/version
   ```

Troubleshooting

To restore system due to Flash memory loss:

If the contents of Flash memory are lost after an upgrade, follow the instructions below to restore your system.

1. Recycle the power on your console server.
2. Using the console, wait for the self test messages.
3. If you get no boot messages, verify that you have the correct setting, otherwise press **s** immediately after turning on the console server to skip an alternate boot code.

   console server boots using its original boot code.

4. During the self test, press **Esc** after the Ethernet test.

   ```
   Testing Ethernet .......................
   ```

5. When the Watch Dog Timer prompt appears, press **Enter**.

   ```
   Watchdog timer ((A)ctive or (I)nactive) [I] :
   ```

6. Choose the option Network Boot when asked.

   ```
   Firmware boot from ((F)lash or (N)etwork) [N] :
   ```

7. Select the TFTP option instead of BootP. The host must be running TFTPD and the new zImage file must be located in the proper directory. For example, /tftpboot for Linux.

   ```
   Boot type ((B)ootp,(T)ftp or Bot(H)) [H] :
   ```

8. Enter the filename of the zImage file on the host.

   ```
   Boot File Name [<zImage_filename>] :
   ```

9. Enter the IP address of the Ethernet interface.

   ```
   IP address assigned to Ethernet interface [192.168.48.11] :
   ```

10. Enter the IP address of the host where the new zImage file is located.

    ```
    Server's IP address [192.168.49.127] :
    ```

11. Accept the default MAC address by pressing **Enter**.

    ```
    MAC address assigned to Ethernet [00:60:2E:01:6B:61] :
    ```

12. When the “Fast Ethernet” prompt appears, press **Enter**.

```
Fast Ethernet ((A)uto Neg, 100 (B)tH, 100 Bt(F), 10 B(t)F, 10 Bt(H))
[A] :
```

The console server should begin to boot off the network and the new image is downloaded. At this point, follow the upgrade process to save the new zImage file into Flash again.

NOTE: Possible causes for the loss of Flash memory may include downloaded wrong zImage file, downloaded as ASCII instead of binary or problems with Flash memory.

If the console server booted properly, the interfaces may be verified using ifconfig and ping. If ping does not work, check the routing table using the command route.

The file /etc/config_files contains a list of files that are affected by saveconf and restoreconf commands. At the command prompt issue the command cat /etc/config_files to see the list of files that are available in the Flash and are loaded into the Ramdisk at the boot time.

NOTE: If any of the files listed in /etc/config_files are modified, the console server administrator must execute the command saveconf before rebooting the console server or the changes are lost. If a file is created (or a filename altered), its name must be added to this file before executing saveconf and rebooting. This speeds up the resolution of most problems.

Setting the maximum number of bytes received by the interface

You can avoid CPU overload by setting a limit to the rate of bytes received. The bootconf utility offers a way of setting this limit. The default is set to 0, which disables the function. For optimum performance set the value to 50000.

To set a limit of bytes received by the interface per second:

1. Run bootconf.

```
Current configuration
MAC address assigned to Ethernet [00:60:2e:00:16:b9]
IP address assigned to Ethernet interface
[192.168.160.10]
Watchdog timer ((A)ctive or (I)nactive) [A]
Firmware boot from ((F)lash or (N)etwork) [F]
Boot type ((B)ootp,(T)ftp or Bot(H)) [T]
Boot File Name [zvmppcts.bin]
Server's IP address [192.168.160.1]
Console speed [9600]
(P)erform or (S)kip Flash test [P]
(S)kip, (Q)uick or (F)ull RAM test [F]
Fast Ethernet ((A)uto Neg, (1)00 BtH, 100 Bt(F), 10
B(t)F, 10 Bt(H)) [A]
Fast Ethernet Maximum Interrupt Events [0]
Maximum rate of incoming bytes per second [0]
```

2. Press **Enter** for all fields but the Maximum rate of incoming bytes per second field.
3. Type the maximum amount of bytes that may be received by the interface per second. A value of zero disables the feature. Enter a value of 50000 for optimum performance.

NOTE: Using larger values does not harm your system but makes it more sensible to storms. Using smaller values may enforce this feature to be triggered by the normal traffic.

4. Save your changes to Flash.

```
Do you confirm these changes in flash ( (Y)es, (N)o (Q)uit ) [N] :
```

LEDs

CPU LEDs

Normally the CPU status LED should blink consistently one second on, one second off. If this is not the case, an error has been detected during the boot. The blink pattern may be interpreted via the following table.

Table B.1: CPU LED Code Interpretation

Event	CPU LED Morse code
Normal Operation	S (short, short, short . . .)
Flash Memory Error - Code	L (long, long, long . . .)

Table B.1: CPU LED Code Interpretation (Continued)

Event	CPU LED Morse code
Flash Memory Error - Configuration	S, L
Ethernet Error	S, S, L
Network Boot Error	S, S, S, S, L
Real-Time Clock Error	S, S, S, S, S, L

NOTE: The Ethernet error mentioned in the previous table occurs automatically if the Fast Ethernet link is not connected to an external hub during the boot. If the Fast Ethernet is not being used or is connected later, this error may be ignored.

Rear panel LEDs

The console server rear panel has serial, console and ethernet connectors with LEDs that have the following functionality.

Ethernet connector

- Col (collision) - Shows collision on the LAN every time the unit tries to transmit an Ethernet packet.
- DT/LK (data transaction/link state) - DT flashes when there is data transmitted to or received from the LAN. It is hardware-controlled. LK keeps steady if the LAN is active. The green LED is Data Transaction activity and the yellow LED is LinK state.
- 100 - If 100BT is detected the LED lights on. If 10BT is detected it turns off.

Console connector

- CP - CPU activity. It flashes at roughly 1 second intervals.
- P1 - Power supply #1 ON.
- P2 - Power supply #2 ON.

NOTE: P1 and P2 LEDs are available only on dual power supply models.

Serial connector

- LK - DTR. It's software-controlled.
- DT - Data transmitted to or received from the serial line. It's hardware-controlled.

Boot configuration

To configure boot parameters:

1. Use the following command to configure the boot parameters of the console server. Refer to Table B.2 for the description of parameters.

   ```
   cli> config administration bootconfig [parameter] <value>
   ```

Table B.2: CLI Boot Configuration Parameters

Parameter	Value	Description
boottype	bootp | both | tftp	To set the network boot type.
bootunit	network	To set from where the unit boots.
consolespeed	115200 | 19200 | 38400 | 4800 | 57600 | 9600	To configure the console speed.
ethernetip	ethernetip | ethernetmode	Assign a temporary IP address to the Ethernet interface.
ethernetmode	100F | 100H | 10F | 10H | auto	To set an Ethernet mode.
filename	<filename>	Add a filename of the image on the tftp server.
flashtest	full | skip	Enable or disable the Flash test.
maxevents	<number>	Set maximum number of Ethernet events handled at once.
ramtest	full | quick | skip	Select a type of ram test.
tftpserver	<n.n.n.n>	Set the IP address of the tftpserver.
wdt	off | on	Enable or disable watch dog timer.

2. Activate and save your configuration.

CLI administration parameters

The administration section of the CLI interface is divided into three parts.

- Session management
- Backup configuration
- Firmware upgrade

Session Management

```
cli> administration sessions [parameter] <value>
```

Table B.3: CLI Session Management Parameters

Parameter	Value	Description
Kill	<Serial Port Number[1-1024]>	To cancel a connection to the serial port <n>
LIst		Lists the current sessions

Backup configuration

Save or restore configuration to an FTP server

```
cli> administration backupconfig [parameter] <value>
```

Table B.4: Backup Configuration Parameters

Parameter Level1	Parameter Level2	Parameter Level3	Value
loadfrom	ftp	username password serverip pathname	<username> <password> <serverip> <pathname>
saveto	ftp	username password serverip pathname	<username> <password> <serverip> <pathname>

In the following example, the command loads a configuration from a server with IP address 192.168.0.1, username john, password john1234 and the configuration file located at /home/configuration.

```
backupconfig> loadfrom serverip 192.168.0.1 pathname
/home/configuration username john password john1234
```

Firmware upgrade

To upgrade the firmware on the console server:

1. Enter the following command at the CLI prompt.

   ```
   cli> administration upgradefw ftpsite <n.n.n.n> username <name>
   password <password> filepathname <path> checksum <yes|no>
   ```

   As an example, the following parameters are used to show the command usage.

   ```
   FTP Server: 192.168.100.111
   Path: /images/zImage
   User: john
   Password: john1234
   ```

   ```
   cli> administration upgradefw ftpsite 192.168.100.111 username john
   password john1234 filepathname /images/zImage checksum no
   ```

2. Activate and save your configuration.
3. Close the CLI session and reboot the console server.

   ```
   cli> quit
   ```

   ```
   # reboot
   ```

Appendix C: Linux File Structure

The Linux file system is organized hierarchically, with the root directory represented by the forward slash (/) symbol. All folders and files are nested within each other below this base directory. Table C.1 displays the Linux directory structure.

Table C.1: Linux Directory Structure

Path	Description
/home	Contains the working directories of the users.
/bin	Contains applications and utilities used during system initialization.
/dev	Contains files for devices and ports.
/etc	Contains configuration files specific to the operating system.
/lib	Contains shared libraries.
/proc	Contains process information.
/mnt	Contains information about mounted disks.
/opt	Location where packages that are not supplied with the operating system are stored.
/tmp	Location where temporary files are stored.
/usr	Contains most of the operating system files.

Basic Linux commands

Table C.2 describe the basic Linux commands for file manipulation or changing directory and contents.

Table C.2: File Manipulation Commands

Command	Description
cp file_name destination	Copies the file indicated by file_name to the path indicated by destination.
• cp text.txt /tmp • cp /chap/robo.php ./excess.php	• Copies the file text.txt in the current directory to the /tmp directory. • Copies the file robo.php in the chap directory to the current directory and renames the copy excess.php.
rm file_name	Removes the file indicated by file_name.
mv file_name destination	Moves the file indicated by file_name to the path indicated by destination.

Table C.2: File Manipulation Commands (Continued)

Command	Description
mkdir directory_name	Creates a directory named directory_name.
• mkdir spot	• Creates the directory spot in the current directory.
• mkdir /tmp/snuggles	• Creates the directory snuggles in the directory /tmp.
rmdir directory_name	Removes the directory indicated by directory_name.
pwd	Supplies the name of the current directory. While logged in, the user is always “in” a directory. The default initial directory is the user's home directory /home/<username>
ls [options] directory_name	Lists the files and directories within directory_name. Some useful options are -l for more detailed output and -a which shows hidden system files.
cd directory_name	Changes the directory to the one specified.
cat file_name	Prints the contents of file_name to the screen.
one dot(.)	Represents the current directory.
two dots (..)	Represents one directory above the current directory.

Appendix D: The vi Editor

To edit a file using the vi editor:

```
# vi file_name
```

The vi editor is a three-state line editor with command, line and editing modes. If in doubt as to which mode you are in, press the **Esc** key, which brings you to the command mode.

Table D.1: vi Modes

Mode	Purpose	How to execute
Command mode	To navigate within an open file.	Press the **Esc** key.
Editing mode	To edit text.	See Table D.2 and Table D.3 for a list of editing commands.
Line mode	To open, save and do other file manipulations.	From the command mode, type colon (**:**)

Use the following keys to navigate to a part of the file you need to edit.

Table D.2: vi Navigation Commands

Command	Description
h	Moves the cursor to the left (left arrow).
j	Moves the cursor to the next line (down arrow).
k	Moves the cursor to the previous line (up arrow).
l	Moves the cursor to the right (right arrow).

Use the following commands to modify the text. Commands -i and -o enforce an edit mode. Press **Esc** to return to the command mode.

Table D.3: vi File Modification Commands

Command	Description
i	Inserts text before the cursor position (everything to the right of the cursor is shifted right).
o	Creates a new line below the current line and insert text (all lines are shifted down).
dd	Removes the entire current line.
x	Deletes the letter at the cursor position.

Once you have completed your file modification, enter the line mode by typing colon (**:**) and one of the following commands.

Table D.4: vi Line Mode Commands

Command	Description
w	Saves the file (w is for write).
wq	Saves and closes the file (q is for quit).
q!	Closes the file without saving.
w file	Saves the file with the name <file>.
e file	Opens the file named <file>.

Appendix E: Technical Support

Our Technical Support staff is ready to assist you with any installation or operating issues you encounter with your Avocent product. If an issue should develop, follow the steps below for the fastest possible service.

To resolve an issue:

1. Check the pertinent section of this manual to see if the issue can be resolved by following the procedures outlined.
2. Visit www.avocent.com/support and use one of the following resources:

   Search the knowledge base or use the online service request.
   -or-
   Select *Technical Support Contacts* to find the Avocent Technical Support location nearest you.

For Technical Support:
www.avocent.com/support

590-814-501B