Cyclades-TS
Installation & Service Manual

Cyclades Corporation
We believe the information in this manual is accurate and reliable. However, we assume no responsibility, financial or otherwise, for any consequences of the use of this product or Installation & Service Manual. This manual is published by Cyclades Corporation, which reserves the right to make improvements or changes in the products described in this manual as well as to revise this publication at any time and without notice to any person of such revision or change. The operating system covered in this manual is V_1.3.3. All brand and product names mentioned in this publication are trademarks or registered trademarks of their respective holders.

**FCC Warning Statement:**
The Cyclades-TS 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 & 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 the Cyclades-TS1000 and the Cyclades-TS2000:**
In order to comply with FCC standards the Cyclades-TS1000 and the Cyclades-TS2000 require the use of a shielded CAT 5 cable for the Ethernet interface. Notice that this cable is not supplied with either of the products and must be provided by the customer.

**Canadian DOC Notice:**
The Cyclades-TS 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.

Le Cyclades-TS n’émet 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.
# Table of Contents

- **CHAPTER 1** HOW TO USE THIS MANUAL ................................................................................................................................. 8
- **CHAPTER 2** SAFETY INSTRUCTIONS ........................................................................................................................................ 9
  - USING YOUR CYCLADES-TS .................................................................................................................................................. 9
  - WORKING INSIDE THE CYCLADES-TS ................................................................................................................................. 10
  - REPLACING THE BATTERY ................................................................................................................................................. 10
- **CHAPTER 3** WHAT IS IN THE BOX ......................................................................................................................................... 11
- **CHAPTER 4** QUICK INSTALLATION GUIDE ............................................................................................................................. 17
  - Configuring using Web .................................................................................................................................................... 17
  - Configuring using Telnet .................................................................................................................................................. 24
- **CHAPTER 5** SUMMARY OF THE CONFIGURATION PROCESS ................................................................................................. 25
- **CHAPTER 6** CONFIGURATION .......................................................................................................................................... 28
  - STEP ONE ........................................................................................................................................................................ 28
  - STEP THREE - CONSOLE SERVER .................................................................................................................................. 30
  - STEP THREE - TERMINAL SERVER .................................................................................................................................. 43
  - STEP THREE - REMOTE ACCESS SERVER .......................................................................................................................... 48
  - STEP FOUR - FOR ALL PROFILES ................................................................................................................................ 55
    - Information applicable only to the Cyclades-TS100 .......................................................................................................... 56
      - Configuring the Cyclades-TS100 for the first time ........................................................................................................ 56
      - Clustering .................................................................................................................................................................. 57
CHAPTER 1 HOW TO USE THIS MANUAL

This manual assumes that the reader understands networking basics and is familiar with the terms and concepts used in Local and Wide Area Networking. The Cyclades-TS is a Linux-based terminal server, which gives it great flexibility. It runs an embedded version of the Linux operating system and Unix and Linux users will find the configuration process very familiar. On the other hand, users not familiar with Unix will have a steeper learning curve, but it is not necessary to be a Unix expert.

Configuration of the equipment is done by editing a few plain-text files (commented sample files for the principal profiles are provided in appendix C), and then updating the versions of the files in the Cyclades-TS. The files can be edited in the Cyclades-TS using the vi editor provided, or in another computer with the environment and text editor of your choice. Unix user or not, we strongly recommend that you follow the steps in this Installation & Service Manual before jumping in.

This manual should be read in the order written, with exceptions given in the text.
CHAPTER 2  SAFETY INSTRUCTIONS

Use the following safety guidelines to protect yourself and your Cyclades-TS.

USING YOUR CYCLADES-TS

CAUTION: Do not operate your Cyclades-TS with the cover removed.

- In order to avoid shorting out your Cyclades-TS when disconnecting the network cable, first unplug the cable from the equipment and then from the network jack. When reconnecting a network cable to the equipment, first plug the cable into the network jack, and then into the equipment.

- To help prevent electric shock, plug the Cyclades-TS into a properly grounded power source. The cable is equipped with a 3-prong plug to help ensure proper grounding. Do not use adapter plugs or remove the grounding prong from the cable. If you have to use an extension cable, use a 3-wire cable with properly grounded plugs.

- To help protect the Cyclades-TS from electrical power fluctuations, use a surge suppressor, line conditioner, or uninterruptible power supply.

- Be sure that nothing rests on the cables of the Cyclades-TS and that they are not located where they can be stepped on or tripped over.

- Do not spill food or liquids on the Cyclades-TS. If it gets wet, contact Cyclades.

- Do not push any objects through the openings of the Cyclades-TS. Doing so can cause fire or electric shock by shorting out interior components.

- Keep your Cyclades-TS away from heat sources and do not block cooling vents.
WORKING INSIDE THE CYCLADES-TS

NOTICE: Do not attempt to service the Cyclades-TS yourself, except following instructions from Cyclades Technical Support personnel. If this is the case, first take the following precautions:

- Turn the Cyclades-TS off.
- Ground yourself by touching an unpainted metal surface on the back of the equipment before touching anything inside it.

REPLACING THE BATTERY

A coin-cell battery maintains date and time information. The TS100 does not have the battery, so the date and time must be kept up to date by ntpclient. If you have to repeatedly reset time and date information after turning on your Cyclades-TS, replace the battery.

CAUTION: A new battery can explode if it is incorrectly installed. Replace the 3 Volt CR2032 battery only with the same or equivalent type recommended by the battery manufacturer. Discard used batteries according to the battery manufacturer’s instructions.
CHAPTER 3 WHAT IS IN THE BOX

The Cyclades-TS is a line of console access and terminal servers. There are several models with differing numbers of serial ports. The following figures show the main units and accessories included in each package and how cables should be connected. The loop-back connector is provided for convenience in case hardware tests are necessary. The RJ-45M - DB-9 F Crossover cable and the RJ-45M - RJ-45 Sun Netra Crossover cable (not shown in the figures) are also included with the TS3000, TS2000, TS1000, TS800 and TS400.

FIGURE 3.1 CYCLADES-TS3000 AND CABLES
FIGURE 3.2 CYCLADES-TS2000 AND CABLES
Chapter 3 - What is in the Box

FIGURE 3.3 CYCLADES-TS1000 AND CABLES
Chapter 3 - What is in the Box

FIGURE 3.4 CYCLADES-TS800 AND CABLES
Chapter 3 - What is in the Box

FIGURE 3.5 CYCLADES-TS400 AND CABLES
Installation Manual

Power Cable

Console Cable

Connect to a COM Serial Port

To Wall Outlet

Installation Manual

Loop-Back Connector

DB-9 Female to DB-25 Male connector

FIGURE 3.6 CYCLADES-TS100 AND CABLES
CHAPTER 4 QUICK INSTALLATION GUIDE

For users familiar with networking, command line interface in Linux or WEB, this chapter gives all the necessary information to quickly configure and start using the Cyclades-TS box. For more detailed information, the next two chapters should be read.

Configuring using Web

The Cyclades-TS box comes with an IP address pre-configured on its Ethernet interface (192.168.160.10). To access that box using your browser please do as follows:

Step 1: From the working station, issue a command to add a route pointing to the network 192.168.160.0 reached through the workstation’s Ethernet interface.

For Linux, the command would be:
```
route add -net 192.168.160.0/24 gw <IP address assigned to the workstation’s Ethernet interface>
```
e.g. if the workstation has IP address 200.246.93.150 the command would be:
```
route add -net 192.168.160.0/24 gw 200.246.93.150
```

For Windows, the command would be:
```
route add 192.168.160.0 mask 255.255.255.0 <IP address assigned to the workstation’s Ethernet interface>
```
e.g. if the workstation has IP address 200.246.93.150 the command would be:
```
rout add 192.168.160.0 mask 255.255.255.0 200.246.93.150
```

Step 2: Point your browser to 192.168.160.10

Step 3: Enter root as login name and tslinux as password

Step 4: Start configuring the parameters presented on the WEB page
WARNING! Type root in the username field and tslinux in the password field to use the Web Configuration Manager. Change the root password as soon as possible: the user database for the Web Configuration Manager is different than the system user database, so the root password can be different.

FIGURE 4.1 LOGIN PAGE OF THE WEB CONFIGURATION MANAGER

After logging in, the screen shown in Figure 4.2 appears.
FIGURE 4.2 PAGE FOLLOWING LOGIN
This page gives a brief description of all menu options.

To change the password:

1. Click on the link Web User Management->Users
2. Select the user root, then click on the Change Password button.
3. Type the new password twice and submit the request.
4. The next page will require a new login, type root and the new password
5. Click on the link Web User Management->Load/Save Configuration and click on the Save Configuration button.
6. Then, click on the link Administration->Load/Save Configuration and click on the Save Configuration to flash button.

To logout, click on the Administration->Log out link.

The General page of the Web Configuration Manager is shown in Fig. 4.3
FIGURE 4.3 GENERAL PAGE OF THE WEB CONFIGURATION MANAGER
A Menu of links is provided along the left side of the page. A summary of what each link leads to is shown in the following figures.

<table>
<thead>
<tr>
<th>Link Name</th>
<th>Description of Page Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Description, Ethernet, DNS, Name Service Access, Data Buffering.</td>
</tr>
<tr>
<td>Syslog</td>
<td>Configuration for the syslog-ng.</td>
</tr>
<tr>
<td>Serial Ports</td>
<td>Configuration for the Portslave package.</td>
</tr>
<tr>
<td>Serial Port Groups</td>
<td>User Groups in Serial Ports Configuration.</td>
</tr>
<tr>
<td>Host Table</td>
<td>Table of hosts in /etc/hosts.</td>
</tr>
<tr>
<td>Static Routes</td>
<td>Static routes defined in /etc/network/st_routes.</td>
</tr>
<tr>
<td>IP Chains</td>
<td>Static Firewall Chains in /etc/network/ipchains.</td>
</tr>
<tr>
<td>Boot Configuration</td>
<td>Configuration of parameters used in the boot process.</td>
</tr>
<tr>
<td>Edit Text File</td>
<td>Tool to read and edit a configuration file.</td>
</tr>
<tr>
<td>System Users</td>
<td>Management of system users defined in /etc/passwd.</td>
</tr>
<tr>
<td>System Groups</td>
<td>Management of system groups defined in /etc/groups.</td>
</tr>
</tbody>
</table>

**FIGURE 4.4 THE CONFIGURATION SECTION**

<table>
<thead>
<tr>
<th>Link Name</th>
<th>Description of Page Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>List of users allowed to access the web server.</td>
</tr>
<tr>
<td>Groups</td>
<td>List of possible access groups.</td>
</tr>
<tr>
<td>Access Limits</td>
<td>List of access limits for specific URL's.</td>
</tr>
<tr>
<td>Load/Save</td>
<td>Load/Save web user configuration in /etc/websum.conf.</td>
</tr>
</tbody>
</table>

**FIGURE 4.5 THE WEB USER MANAGEMENT SECTION**
<table>
<thead>
<tr>
<th>Link Name</th>
<th>Description of Page Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logout</td>
<td>Exits the Web Manager.</td>
</tr>
<tr>
<td>Reboot</td>
<td>Resets the equipment.</td>
</tr>
<tr>
<td>Port Conversation</td>
<td>Does a port conversation through a serial port.</td>
</tr>
<tr>
<td>Download/Upload Image</td>
<td>Uses an FTP server to load and save a kernel image.</td>
</tr>
<tr>
<td>Load/Save Configuration</td>
<td>Uses flash memory or an FTP server to load or save the TS's configuration.</td>
</tr>
<tr>
<td>Set Date/Time</td>
<td>Set the TS's date and time.</td>
</tr>
<tr>
<td>Active Sessions</td>
<td>Shows the active sessions and allows the administrator to kill them.</td>
</tr>
<tr>
<td>Process Status</td>
<td>Shows the running processes and allows the administrator to kill them.</td>
</tr>
<tr>
<td>Restart Processes</td>
<td>Allows the administrator to start or stop some processes.</td>
</tr>
</tbody>
</table>

**FIGURE 4.6 THE ADMINISTRATION SECTION**

<table>
<thead>
<tr>
<th>Link Name</th>
<th>Description of Page Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Statistics</td>
<td>Shows statistics for all active interfaces.</td>
</tr>
<tr>
<td>DHCP client</td>
<td>Shows the DHCP client information.</td>
</tr>
<tr>
<td>Serial Ports</td>
<td>Shows the status of all serial ports</td>
</tr>
<tr>
<td>Routing Table</td>
<td>Shows the routing table and allows the administrator to add or delete routes.</td>
</tr>
<tr>
<td>ARP Cache</td>
<td>Shows the ARP cache.</td>
</tr>
<tr>
<td>IP Chains</td>
<td>Shows IP Chains Entries.</td>
</tr>
<tr>
<td>IP Rules</td>
<td>Shows Firewall, NAT and IP Accounting rules.</td>
</tr>
<tr>
<td>IP Statistics</td>
<td>Shows IP protocol statistics.</td>
</tr>
<tr>
<td>ICMP Statistics</td>
<td>Shows ICMP protocol statistics.</td>
</tr>
<tr>
<td>TCP Statistics</td>
<td>Shows TCP protocol statistics.</td>
</tr>
<tr>
<td>UDP Statistics</td>
<td>Shows UDP protocol statistics.</td>
</tr>
<tr>
<td>RAM Disk Usage</td>
<td>Shows the TS File System.</td>
</tr>
<tr>
<td>System Information</td>
<td>Shows information about the kernel, Time, CPU and Memory.</td>
</tr>
</tbody>
</table>

**FIGURE 4.7 THE INFORMATION SECTION**
Configuring using Telnet

The Cyclades-TS box comes with an IP address pre-configured on its Ethernet interface (192.168.160.10). To access that box using telnet please do as follows:

**Step 1:** From the working station, issue a command to add a route pointing to the network 192.168.160.0 reached through the workstation’s Ethernet interface.

For Linux, the command would be:

```
route add -net 192.168.160.0/24 gw <IP address assigned to the workstation’s Ethernet interface>
```

**e.g.** if the workstation has IP address 200.246.93.150 the command would be:

```
route add -net 192.168.160.0/24 gw 200.246.93.150
```

For Windows, the command would be:

```
route add 192.168.160.0 mask 255.255.255.0 <IP address assigned to the workstation’s Ethernet interface>
```

**e.g.** if the workstation has IP address 200.246.93.150 the command would be:

```
route add 192.168.160.0 mask 255.255.255.0 200.246.93.150
```

**Step 2:** `telnet 192.168.160.10`

**Step 3:** Enter `root` as login name and `tslinux` as password

**NOTE:** Now, to configure the basic parameters for the Cyclades-TS, type “wiz” at the command prompt. This will start the wizard configuration application, which can be run at any time from the command prompt. In future firmware releases, more functions will be added to the wizard to allow advanced mode configuration be performed using wizard. (See Appendix F Configuration Wizard for more information).
The Cyclades-TS can be used as a:

- console server,
- terminal server,
- remote access server.

A detailed description of each of these profiles is provided in the next chapter. The Cyclades-TS’s operating system is embedded Linux. Even if you are a Unix user and find the tools and files familiar, do not configure this product as you would configure a regular Linux server.

You do not need to be a Unix user to configure the Cyclades-TS. Additional information about the files and tools needed for configuration is provided in appendix A.

The basic configuration steps are:

A. Connecting the Cyclades-TS to the network and other devices. Consult Chapter 3, What is in the Box, for questions on which cable should be used for which device.

B. Connect a PC or terminal to the Cyclades-TS via the console port and login.

C. Modify the Linux following files to let the Cyclades-TS know about its local environment:
   
   /etc/hostname
   /etc/TIMEZONE (see Appendix K for more information)
   /etc/hosts
   /etc/resolv.conf
   /etc/network/st_routes
   /etc/inittab (Cyclades-TS100 only. See “Configuring the Cyclades-TS100 for the First Time” in chapter 6)
D. Change password for root and new users.

The default /etc/passwd file has the user “root” with password “tslinux”. The customer should change the password for user root as soon as possible. Before changing any password or adding new users the customer should also activate shadow password, if it is needed. The Cyclades-TS has support for shadow password, but it is not active by default. To activate shadow password follow the steps listed below:

1. Create an empty file /etc/shadow

    # cd /etc
    # touch shadow

2. Add a temporary user to the system, it will be removed later.

    # adduser boo

3. Edit the file shadow. For each user in passwd file, create a copy of the line that begins with “boo:” in the shadow file, then replace “boo” with the user name. The root’s line must be the first one.

4. Edit the passwd file and replace all fields password with “x”. The root’s line will look like:

    “root:x:0:0:root:/root:/bin/sh”

    password field

TIP: Using the vi editor, put the cursor in the first byte after “root:”, then type “ct:x” plus <ESC>.
5. Remove the temporary user boo.
   
   # deluser boo

6. Change the password for all users and add the new ones needed.

   # passwd <username>
   or
   # adduser <username>

7. Edit config_files file and add a line with “/etc/shadow”.

E. Edit the pslave.conf file. This is the main configuration file that concentrates most product parameters and defines the functionality of the Cyclades-TS. The modifications made to this file will depend on the profile.

F. Activate the changes.

G. Test the configuration to make sure the ports have been set up properly.

H. Save the changes and restart the server application.

Full details on each step listed above and how to perform them are provided in the next chapter. Make sure to always complete ALL the steps for your application before testing or switching to another profile.

WARNING! The Cyclades-TS provides both a command-line and a web interface for your convenience. Both are enabled by default and both have default passwords. Make sure BOTH default passwords (password is tslinux) are changed to avoid unauthorized access to your network. To disable the WEB service, refer to Appendix D.
CHAPTER 6  CONFIGURATION

This chapter guides you step by step through the configuration of the Cyclades-TS for the three principal applications:

1. Console Server,
2. Terminal Server, and

Many steps are common to both, so please read the entire chapter before beginning.

STEP ONE

Connect a PC or terminal to the Cyclades-TS using the console cable. If using a PC, HyperTerminal can be used in the Windows operating system and Kermit or Minicom in the Unix operating system. The terminal parameters should be set as follows:

- Serial Speed: 9600 bps
- Data Length: 8 bits
- Parity: None
- Stop Bits: 1 stop bit
- Flow Control: none
- Ansi emulation (Note: if your terminal does not have ansi emulation, select vt100; then, on the TS, log in as root and switch to vt100 by typing “TERM=vt100;export TERM”)

When the Cyclades-TS boots properly, a login banner will appear.

Log in as root (default password is tslinux). A new password should be created as soon as possible. The Cyclades-TS runs Linux, a Unix-like operating system, and those familiar with the Unix operating system will feel quite at home. A description of the Linux file system and basic commands is given in Appendix A at the end of this manual.
STEP TWO

Any configuration change must be saved in flash once validated. To save in flash run saveconf (seen later in this chapter). To validate a configuration run signal_ras hup and check for the ending results (seen later in this chapter).

In this step, four Linux files must be modified to identify the Cyclades-TS and its neighbors. Then, the boot parameters are configured. The operating system provides the vi editor, which is described in the Linux appendix for the uninitiated. The first file is /etc/hostname. The only entry should be the hostname of the Cyclades-TS. An example is shown in Figure 6.1.

```
TS1000
```

*FIGURE 6.1 CONTENTS OF THE /ETC/HOSTNAME FILE*

The second file is /etc/hosts. It should contain the IP address for the Ethernet interface and the same hostname entered in the /etc/hostname file. It may also contain IP addresses and host names for other hosts in the network.

```
200.200.200.1 TS1000
200.200.200.2 RadiusServer
127.0.0.1 localhost
```

*FIGURE 6.2 CONTENTS OF THE /ETC/HOSTS FILE*

The third file that must be modified is /etc/resolv.conf. It must contain the domain name and nameserver information for the network.

```
domain mycompany.com
nameserver 200.200.200.2
```

*FIGURE 6.3 CONTENTS OF THE /ETC/RESOLV.CONF FILE*
The fourth file defines static routes and is called /etc/network/st_routes. In the console server example in Figure 6.5, the PR1000 is the gateway router and thus its IP address is configured in this file to be the default gateway. Other static routes are also configured in this file.

```bash
route add default gw 200.200.200.5
```

**FIGURE 6.4 CONTENTS OF THE /ETC/NETWORK/ST_ROUTES FILE**

NOTE: We strongly recommend to use 9600 bps console speed. In case you need to use other speed please check the troubleshooting session.

**STEP THREE**
This is where the configuration for the three profiles - Console Server, Terminal Server and Remote Access Server diverge. Follow step three for the appropriate profile.

**STEP THREE - CONSOLE SERVER**
A console server application is shown in Figure 6.5.
This application allows a user to access a server connected to the Cyclades-TS through its serial console port from a workstation on the LAN or WAN. A server console is opened on the workstation. The authentication is usually performed by a Radius server and either telnet or ssh (a secure shell session) can be used. See the Linux appendix for more information about ssh.

The fifth file is specific to the Cyclades-TS and a sample file with comments is supplied in the Linux file system. It is called /etc/portslave/pslave.conf. A listing of pslave.conf with all possible parameters, as well as sample files used to create the three applications in this chapter, is provided in Appendix C. There are three basic types of
parameters: conf.* parameters are global or apply to the Ethernet interface; all.* parameters are used to set default parameters for all ports, and s#./* parameters change the default port parameters for individual ports. An all.* parameter can be overridden by a s#./* parameter appearing later in the pslave.conf file (or vice-versa). A brief description of each parameter used for the console server profile is given in Figures 6.6-6.7.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf.eth_ip</td>
<td>The IP address of the Ethernet interface. This parameter, along with the next two, is used by the cy_ras program to OVERWRITE the file /etc/network/ifcfg_eth0 as soon as the command &quot;signal_ras hup&quot; is executed. The file /etc/network/ifcfg_eth0 should not be edited by the user unless the cy_ras application is not going to be used.</td>
<td>200.200.200.1</td>
</tr>
<tr>
<td>conf.eth_mask</td>
<td>The mask for the Ethernet network.</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>conf.eth_mtu</td>
<td>The Maximum Transmission Unit size, which determines whether or not packets should be broken up.</td>
<td>1500</td>
</tr>
<tr>
<td>conf.nfs_data_buffering</td>
<td>Remote Network File System where data captured from the serial port will be written instead of the default directory &quot;/var/run/DB&quot;. The directory tree to which the file will be written must be NFS-mounted. If data buffering is turned on for port 1, for example, the data will be stored in the file ttyS1.data (or &lt;serverfarm1&gt;.data if s1.serverfarm was configured) in the directory indicated by this variable (please see also Data Buffering section for more details). The remote host must have NFS installed and the administrator must create, export and allow reading/writing to this 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 size of the file is dependent on the NFS server only (hard drive, partition size, etc.).</td>
<td>commented</td>
</tr>
<tr>
<td>conf.lockdir</td>
<td>The lock directory, which is /var/lock for the Cyclades-TS. It should not be changed unless the user decides to customize the operating system.</td>
<td>/var/lock</td>
</tr>
</tbody>
</table>

*FIGURE 6.6 CONSOLE SERVER PSLAVE.CONF GLOBAL PARAMETERS*
### Parameter Description Value

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value in Exp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf.facility</td>
<td>This value (0-7) is the Local facility sent to the syslog. The file /etc/syslog-ng/syslog-ng.conf contains a mapping between the facility number and the action (see more in Appendix G).</td>
<td>7</td>
</tr>
<tr>
<td>conf.DB_facility</td>
<td>This value (0-7) is the Local facility sent to the syslog with the data when syslog_buffering and/or alarm are active. The file /etc/syslog-ng/syslog-ng.conf contains a mapping between the facility number and the action (see more in Appendix G).</td>
<td>0</td>
</tr>
<tr>
<td>conf.group</td>
<td>Used to group users to simplify configuration of the parameter all.users later on. This parameter can be used to define more than one group.</td>
<td>group_name: user1, user2</td>
</tr>
<tr>
<td>all.speed</td>
<td>The speed for all ports. This value (as for any &quot;all.&quot; parameters) can later be overridden for individual ports using the $&lt;port number&gt;.speed$ parameter.</td>
<td>9600</td>
</tr>
<tr>
<td>all.datasize</td>
<td>The data size for all ports.</td>
<td>8</td>
</tr>
<tr>
<td>all.stopbits</td>
<td>The number of stop bits for all ports</td>
<td>1</td>
</tr>
<tr>
<td>all.parity</td>
<td>The parity for all ports.</td>
<td>none</td>
</tr>
<tr>
<td>all.dcd</td>
<td>DCD signal (sets the tty parameter CLOCAL). Valid values are 0 or 1. In a socket session, if all.dcd=0, a connection request (telnet or ssh) will be accepted regardless of the DCD signal and the connection and will not be closed if the DCD signal is set to DOWN. In a socket connection, if all.dcd=1 a connection request will be accepted only if the DCD signal is UP and the connection (telnet or ssh) will be closed if the DCD signal is set to DOWN.</td>
<td>0</td>
</tr>
<tr>
<td>all.modbus_smode</td>
<td>Communication mode through the serial ports. This parameter is meaningful only when modbus protocol is configured. The valid options are ascii (normal TX/RX mode) and rtu (some time constraints are observed between characters while transmitting a frame). If not configured, ASCII mode will be assumed.</td>
<td>commented</td>
</tr>
</tbody>
</table>

**FIGURE 6.6 CONSOLE SERVER PSLAVE.CONF GLOBAL PARAMETERS (CONT.)**

**FIGURE 6.7 CONSOLE SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS**
### Parameter | Description | Value for This Example
--- | --- | ---
all.authtype | There are several authentication type options: local (authentication is performed using the /etc/passwd file), radius (authentication is performed using a Radius authentication server), TacacsPlus (authentication is performed using a TacacsPlus authentication server), none, local/radius (authentication is performed locally first, switching to Radius if unsuccessful), radius/local (the opposite of the previous option), RadiusDownLocal (local authentication is tried only when the Radius server is down), local/TacacsPlus (authentication is performed locally first, switching to TacacsPlus if unsuccessful), TacacsPlus/local (the opposite of the previous option), TacacsPlusDownLocal (local authentication is tried only when the TacacsPlus server is down). Note that this parameter controls the authentication required by the Cyclades-TS. The authentication required by the device to which the user is connecting is controlled separately. | radius
all.authhost1 | This address indicates the location of the Radius/TacacsPlus authentication server and is only necessary if this option is chosen in the previous parameter. A second Radius/TacacsPlus authentication server can be configured with the parameter all.authhost2. | 200.200.200.2
all.accthost1 | This address indicates the location of the Radius/TacacsPlus accounting server, which can 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 will not be performed. If the same server is used for authentication and accounting, both parameters must be filled with the same address. A second Radius/TacacsPlus accounting server can be configured with the parameter all.accthost2. | 200.200.200.2

*FIGURE 6.7 CONSOLE SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONT.)*
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.radtimeout</td>
<td>This is the timeout (in seconds) for a Radius/TacacsPlus authentication query to be answered. The first server (authhost1) is tried &quot;radretries&quot; times, and then the second (authhost2), if configured, is contacted &quot;radretries&quot; times. If the second also fails to respond, Radius/TacacsPlus authentication fails.</td>
<td>3</td>
</tr>
<tr>
<td>all.radretries</td>
<td>Defines the number of times each Radius/TacacsPlus server is tried before another is contacted. The default, if not configured, is 5.</td>
<td>5</td>
</tr>
<tr>
<td>all.secret</td>
<td>This is the shared secret necessary for communication between the Cyclades-TS and the Radius/TacacsPlus servers.</td>
<td>cyclades</td>
</tr>
<tr>
<td>all.ipno</td>
<td>This is the default IP address of the Cyclades-TS's serial ports. The &quot;+&quot; indicates that the first port should be addressed as 192.168.1.101 and the following ports should have consecutive values. Any host can access a port using its IP address as long as a path to the address exists in the host's routing table.</td>
<td>192.168.1.101+</td>
</tr>
<tr>
<td>all.issue</td>
<td>This text determines the format of the login banner that is issued when a connection is made to the Cyclades-TS. \n represents a new line and \r represents a carriage return. Expansion characters, listed in Appendix C, can be used here.</td>
<td>\n\n\nTSLINUX - Portslave Internet Services\n\n\nWelcome to terminal server %h port S%p \n\nCustomer Support: 510-770-9727 <a href="http://www.cyclades.com%5Cn%5Cn%5Cn">www.cyclades.com\n\n\n</a></td>
</tr>
<tr>
<td>all.prompt</td>
<td>This text defines the format of the login prompt. Expansion characters, listed in Appendix C, can be used here.</td>
<td>%h login:</td>
</tr>
<tr>
<td>all.flow</td>
<td>This sets the flow control to hardware, software, or none.</td>
<td>hard</td>
</tr>
</tbody>
</table>

*FIGURE 6.7  CONSOLE SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONT.)*
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.poll_interval</td>
<td>Valid only for protocols socket_server, socket_ssh and raw_data. When not set to zero, this parameter sets the wait for a TCP connection keep-alive timer (in milliseconds). If no traffic passes through the Cyclades-TS for this period of time, the Cyclades-TS will send a line status message to the remote device to see if the connection is still up. If not configured, 1000 ms is assumed. If set to zero, line status messages will not be sent to the socket client.</td>
<td>0</td>
</tr>
<tr>
<td>all.socket_port</td>
<td>This defines an alternative labeling system for the Cyclades-TS ports. The '+' after the numerical value causes the interfaces to be numbered consecutively. In this example, interface 1 is assigned the port value 7001, interface 2 is assigned the port value 7002, etc.</td>
<td>7001+</td>
</tr>
<tr>
<td>all.protocol</td>
<td>For the console server profile, the possible protocols are socket_server (when telnet is used), socket_ssh (when ssh version one or two is used), raw_data (to exchange data in transparent mode – similar to socket_server mode, but without telnet negotiation, breaks to serial ports, etc.), or modbus (an application layer messaging protocol for client/server communication widely used for industrial automation – see Appendix I for details on Modbus protocol).</td>
<td>socket_server</td>
</tr>
</tbody>
</table>

**FIGURE 6.7  CONSOLE SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONT.)**
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.data_buffering</td>
<td>A non zero value activates data buffering (local or remote, according to what was configured in the parameter conf.nfs_data_buffering seen before). If local data buffering, a file is created on the Cyclades-TS; if remote, a file is created through NFS in a remote server. All data received from the port is captured in this file. If local data buffering, this parameter means the maximum file size (in bytes). If remote, this parameter is just a flag to activate (greater than zero) or deactivate data buffering. When local data buffering is used, each time the maximum is reached the oldest 10% of stored data is discarded, releasing space for new data (FIFO system) - circular file. When remote data buffering is used, there's no maximum file size other than the one imposed by the remote server - linear file. This file can be viewed using the normal Unix tools (cat, vi, more, etc.). See the section on data buffering for details.</td>
<td>0</td>
</tr>
<tr>
<td>all.DB_timestamp</td>
<td>A non zero value activates time stamp recording in the data buffering file. This parameter is meaningful only if data buffering option is active. In case time stamp recording is on, input characters will be accumulated until either a CR or LF character is received from the serial port or the size of the accumulated data reaches 256 characters. Then, the accumulated data will be recorded in the data buffering file along with the current time.</td>
<td>0</td>
</tr>
<tr>
<td>all.syslog_buffering</td>
<td>When non zero, the contents of the data buffer are sent to the syslog-ng every time a quantity of data equal to this parameter is collected. The syslog level for data buffering is hard coded to level 5 (notice) and facility conf.DB_facility. The file /etc/syslog-ng/syslog-ng.conf should be set accordingly for the syslog-ng to take some action (please see Appendix G for syslog-ng configuration file).</td>
<td>0</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value for This Example</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>all.dont_show_DB menu</td>
<td>When zero, a menu with data buffering options is shown when a nonempty data buffering file is found. When 1, the data buffering menu is not shown. When 2, the data buffering menu is not shown but the data buffering file is shown if not empty. When 3, the data buffering menu is shown, but without the erase and show and erase options.</td>
<td>1</td>
</tr>
<tr>
<td>all.alarm</td>
<td>When non zero, all data received from the port are captured and sent to syslog-ng with LOCAL [0+DB_facility] facility and INFO level. The file /etc/syslog-ng/syslog-ng.conf should be set accordingly, for the syslog-ng to take some action (please see Appendix G for syslog-ng configuration file).</td>
<td>0</td>
</tr>
<tr>
<td>all.users</td>
<td>Restricts access to ports by user name (only the users listed can access the port or, using the character &quot;!&quot;, all but the users listed can access the port.) In this example, the users joe, mark and members of user_group cannot access the port. A single comma and spaces/tabs may be used between names. A comma may not appear between the ! and the first user name. The users may be local, Radius or TacacsPlus. User groups (defined with the parameter conf.group) can be used in combination with user names in the parameter list. Notice that these are common users, not administrators.</td>
<td>! joe, mark, user_group</td>
</tr>
<tr>
<td>all.sniff_mode</td>
<td>This parameter determines what other users connected to the very same port (see parameter admin_users below) can see of the session of the first connected user (main session): in shows data written to the port, out shows data received from the port, and i/o shows both streams. The second and later sessions are called sniff sessions and this feature may be activated only when the protocol parameter is set to socket_ssh or socket_server.</td>
<td>out</td>
</tr>
</tbody>
</table>

**FIGURE 6.7  CONSOLE SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONT.)**
Table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.admin_users</td>
<td>This parameter determines which users can open a sniff session, which is where other users connected to the very same port can see everything that a first user connected is doing. The other users connected to the very same port can also cancel the first user’s session (and take over). If all.multiple_sessions (seen below) is configured as no only two users can connect to the same port simultaneously. If all.multiple_sessions is configured as yes more simultaneous users can sniff the session or have read and/or write permission (please see details in Appendix E). When users want access per port to be controlled by administrators, this parameter is obligatory and authtype must not be none. This parameter can determine who can open a sniff session or cancel a previous session. User groups (defined with the parameter conf.group) can be used in combination with user names in the parameter list.</td>
<td>peter, john, user_group</td>
</tr>
<tr>
<td>all.multiple_sessions</td>
<td>Valid for all serial ports; must be “yes” or “no”. If it is not defined, the default will be “no”. Please see Appendix E for details.</td>
<td>no</td>
</tr>
<tr>
<td>all.escape_char</td>
<td>Valid for all the serial ports with session sniffing enabled (all.admin_users); this parameter will be used to present the menus to the user. The format of this parameter will be set as “^x”, where x is the keystroke of the escape character. Only characters from “^a” to “^z” (i.e. CTRL-A to CTRL-Z) will be accepted. The default value is “^z”. Please see Appendix E for details.</td>
<td>^z</td>
</tr>
<tr>
<td>all.tx_interval</td>
<td>Valid for protocols socket_server, socket_ssh and raw_data. Defines the delay (in milliseconds) before transmission to the Ethernet of data received through a serial port. If not configured, 100ms is assumed. If set to zero or a value above 1000, no buffering will take place.</td>
<td>100</td>
</tr>
</tbody>
</table>

FIGURE 6.7 CONSOLE SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONT.)
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.idletimeout</td>
<td>Valid only for the CAS profile (protocols socket_server, socket_ssh, raw_data and modbus). Specifies how long (in minutes) a connection can remain inactive before it is cut off. If set to zero (the default), the connection will not time out.</td>
<td>0</td>
</tr>
<tr>
<td>all.sttyCmd</td>
<td>Tty settings after a socket connection to that serial port is established. The tty is programmed to work as a CAS profile and this user specific configuration is applied over that serial port. Parameters must be separated by space. (e.g., the following example sets -igncr which tells the terminal not to ignore the carriage-return on input, -onlcr do not map newline character to a carriage return/newline character sequence on output, opost post-process output, -icrnl do not map carriage-return to a newline character on input. all.sttyCmd -igncr -onlcr opost -icrnl)</td>
<td>commented</td>
</tr>
<tr>
<td>s1.tty</td>
<td>The device name for the port is set to the value given in this parameter. If a device name is not provided for a port, it will not function.</td>
<td>ttyS1</td>
</tr>
</tbody>
</table>

6.7 CONSOLE SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONT.)
There are several authentication type options: local (authentication is performed using the /etc/passwd file), radius (authentication is performed using a Radius authentication server), TacacsPlus (authentication is performed using a TacacsPlus authentication server), none, local/radius (authentication is performed locally first, switching to Radius if unsuccessful), radius/local (the opposite of the previous option), RadiusDownLocal (local authentication is tried only when the Radius server is down), local/TacacsPlus (authentication is performed locally first, switching to TacacsPlus if unsuccessful), TacacsPlus/local (the opposite of the previous option), TacacsPlusDownLocal (local authentication is tried only when the TacacsPlus server is down). Note that this parameter controls the authentication required by the Cyclades-TS. The authentication required by the device to which the user is connecting is controlled separately.

Note: if the sniff session feature is used for a specific port, authtype parameter must not be set to none. If none is chosen, any user can open a sniff session and/or cancel sessions of other users for this port.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1.authtype</td>
<td>There are several authentication type options: local (authentication is performed using the /etc/passwd file), radius (authentication is performed using a Radius authentication server), TacacsPlus (authentication is performed using a TacacsPlus authentication server), none, local/radius (authentication is performed locally first, switching to Radius if unsuccessful), radius/local (the opposite of the previous option), RadiusDownLocal (local authentication is tried only when the Radius server is down), local/TacacsPlus (authentication is performed locally first, switching to TacacsPlus if unsuccessful), TacacsPlus/local (the opposite of the previous option), TacacsPlusDownLocal (local authentication is tried only when the TacacsPlus server is down). Note that this parameter controls the authentication required by the Cyclades-TS. The authentication required by the device to which the user is connecting is controlled separately. Note: if the sniff session feature is used for a specific port, authtype parameter must not be set to none. If none is chosen, any user can open a sniff session and/or cancel sessions of other users for this port.</td>
<td>local</td>
</tr>
<tr>
<td>s1.serverfarm</td>
<td>Alias name given to the server connected to the serial port.</td>
<td>Server_connected_serial1</td>
</tr>
<tr>
<td>s2.tty</td>
<td>See the s1.tty entry in this table.</td>
<td>ttyS2</td>
</tr>
<tr>
<td>s8.tty</td>
<td>See the s1.tty entry in this table.</td>
<td>ttyS8</td>
</tr>
</tbody>
</table>

6.7 CONSOLE SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONT.)

Execute the command `signal_ras hup` to activate the changes. At this point, the configuration should be tested. A step-by-step check list follows.

1. Since Radius authentication was chosen, create a new user on the Radius authentication server called test and provide him with the password test.
2. From the console, ping 200.200.200.2 to make sure the Radius authentication server is reachable.
3. Make sure that the physical connection between the Cyclades-TS and the servers is correct. A cross cable (not the modem cable provided with the product) should be used. Please see the hardware specifications appendix for pin-out diagrams.
4. The Cyclades-TS has been set for communication at 9600 bps, 8N1. The server must also be configured to communicate on the serial console port with the same parameters. Also make sure that the computer is configured to route console data to its serial console port (Console Redirection).
5. From a server on the LAN (not from the console), try to telnet to the server connected to the first port of the Cyclades-TS using the following command:

   \texttt{telnet 200.200.200.1 7001}

For both telnet and ssh sessions, the servers can be reached by either:

1. Ethernet IP of the Cyclades-TS and assigned socket port
   or
2. Individual IP assigned to each port.

If everything is configured correctly, a telnet session should open on the server connected to port 1. If not, check the configuration, follow the steps above again, and check the troubleshooting appendix. Now continue on to step four later in this chapter.

\textbf{NOTE:} It is possible to access the serial ports from Microsoft stations using some off-the-shelf packages. Although Cyclades is not liable for those packages, successful tests were done using at least one of them. From the application’s viewpoint running on a Microsoft station, the remote serial port works like a regular COM port. All the I/O with the serial device attached to the Cyclades-TS is done through socket connections opened by these packages and a COM port is emulated to the application.
STEP THREE - TERMINAL SERVER

The terminal server profile allows a terminal user to access a server on the LAN. The terminal can be either a dumb terminal or a terminal emulation program on a PC. No authentication is used in this example and rlogin is chosen as the protocol.

The fifth configuration file (the first four were described in step two) is specific to the Cyclades-TS and a sample file with comments is supplied in the Linux file system. It is called /etc/portslave/pslave.conf. A listing of pslave.conf with all possible parameters, as well as sample files used to create the three applications in this chapter, is provided in Appendix C. There are three basic types of parameters: conf.* parameters are global or apply to the Ethernet interface; all.* parameters are used to set default parameters for all ports, and s#. * parameters change the default port parameters for individual ports. An all.* parameter can be overridden by a s#.* parameter appearing...
later in the pslave.conf file (or vice-versa). A brief description of each parameter used for the terminal server profile is given in Figures 6.9-6.10.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf.eth_ip</td>
<td>The IP address of the Ethernet interface. This parameter, along with the next two, is used by the cy_ras program to OVERWRITE the file /etc/network/ifcfg_eth0 as soon as the command &quot;signal_ras hup&quot; is executed. The file /etc/network/ifcfg_eth0 should not be edited by the user unless the cy_ras application is not going to be used.</td>
<td>200.200.200.1</td>
</tr>
<tr>
<td>conf.eth_mask</td>
<td>The mask for the Ethernet network.</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>conf.eth_mtu</td>
<td>The Maximum Transmission Unit size, which determines whether or not packets should be broken up.</td>
<td>1500</td>
</tr>
<tr>
<td>conf.lockdir</td>
<td>The lock directory, which is /var/lock for the Cyclades-TS. It should not be changed unless the user decides to customize the operating system.</td>
<td>/var/lock</td>
</tr>
<tr>
<td>conf.rlogin</td>
<td>Location of the rlogin binary that accepts the -i flag.</td>
<td>/usr/local/bin/rlogin-radius</td>
</tr>
<tr>
<td>conf.telnet</td>
<td>Location of the telnet utility.</td>
<td>/bin/telnet</td>
</tr>
<tr>
<td>conf.ssh</td>
<td>Location of the ssh utility.</td>
<td>/bin/ssh</td>
</tr>
<tr>
<td>conf.locallogins</td>
<td>This parameter is only necessary when authentication is being performed for a port. When set to one, it is possible to log in to the Cyclades-TS directly by placing a &quot;!&quot; before your login name, then using your normal password. This is useful if the Radius authentication server is down.</td>
<td>0</td>
</tr>
</tbody>
</table>

*FIGURE 6.9 TERMINAL SERVER PSLAVE.CONF GLOBAL PARAMETERS*
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.speed</td>
<td>The speed for all ports. <strong>This value (as for any &quot;all.&quot; parameters) can later be overridden for individual ports using the s&lt;port number&gt;.speed parameter.</strong></td>
<td>9600</td>
</tr>
<tr>
<td>all.datasize</td>
<td>The data size for all ports.</td>
<td>8</td>
</tr>
<tr>
<td>all.stopbits</td>
<td>The number of stop bits for all ports</td>
<td>1</td>
</tr>
<tr>
<td>all.parity</td>
<td>The parity for all ports.</td>
<td>none</td>
</tr>
<tr>
<td>all.dcd</td>
<td>DCD signal (sets the tty parameter CLOCAL). Valid values are 0 or 1. In a socket session, if all.dcd=0, a connection request (telnet or ssh) will be accepted regardless of the DCD signal and the connection and will not be closed if the DCD signal is set to DOWN. In a socket connection, if all.dcd=1 a connection request will be accepted only if the DCD signal is UP and the connection (telnet or ssh) will be closed if the DCD signal is set to DOWN.</td>
<td>0</td>
</tr>
<tr>
<td>all.authtype</td>
<td>There are several authentication type options: local (authentication is performed using the /etc/passwd file), radius (authentication is performed using a Radius authentication server), TacacsPlus (authentication is performed using a TacacsPlus authentication server), none, local/radius (authentication is performed locally first, switching to Radius if unsuccessful), radius/local (the opposite of the previous option), RadiusDownLocal (local authentication is tried only when the Radius server is down), local/TacacsPlus (authentication is performed locally first, switching to TacacsPlus if unsuccessful), TacacsPlus/local (the opposite of the previous option), TacacsPlusDownLocal (local authentication is tried only when the TacacsPlus server is down). Note that this parameter controls the authentication required by the Cyclades-TS. The authentication required by the device to which the user is connecting is controlled separately.</td>
<td>none</td>
</tr>
</tbody>
</table>

**FIGURE 6.10 TERMINAL SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS**
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.authhost1</td>
<td>This address indicates the location of the Radius/TacacsPlus authentication server and is only necessary if this option is chosen in the previous parameter. A second Radius/TacacsPlus authentication server can be configured with the parameter all.authhost2.</td>
<td>200.200.200.2</td>
</tr>
<tr>
<td>all.accthost1</td>
<td>This address indicates the location of the Radius/TacacsPlus accounting server, which can 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 will not be performed. If the same server is used for authentication and accounting, both parameters must be filled with the same address. A second Radius/TacacsPlus accounting server can be configured with the parameter all.accthost2.</td>
<td>200.200.200.2</td>
</tr>
<tr>
<td>all.radtimeout</td>
<td>This is the timeout (in seconds) for a Radius/TacacsPlus authentication query to be answered. The first server (authhost1) is tried &quot;radretries&quot; times, and then the second (authhost2), if configured, is contacted &quot;radretries&quot; times. If the second also fails to respond, Radius/TacacsPlus authentication fails.</td>
<td>3</td>
</tr>
<tr>
<td>all.radretries</td>
<td>Defines the number of times each Radius/TacacsPlus server is tried before another is contacted. The default, if not configured, is 5.</td>
<td>5</td>
</tr>
<tr>
<td>all.secret</td>
<td>This is the shared secret necessary for communication between the Cyclades-TS and the Radius/TacacsPlus servers.</td>
<td>cyclades</td>
</tr>
<tr>
<td>all.protocol</td>
<td>For the terminal server profile, the possible protocols are login (which requests username and password), rlogin (which receives the username from the TS and requests a password), telnet, socket_client, ssh and ssh2.</td>
<td>rlogin</td>
</tr>
<tr>
<td>all.host</td>
<td>The IP address of the host to which the terminals will connect.</td>
<td>200.200.200.3</td>
</tr>
</tbody>
</table>

FIGURE 6.10 TERMINAL SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONTINUED)
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.issue</td>
<td>This text determines the format of the login banner that is issued when a connection is made to the Cyclades-TS. \n represents a new line and \r represents a carriage return.</td>
<td>\n\n\nTSLINUX - Portslave Internet Services\n\nWelcome to terminal server %h port S%p \n\nCustomer Support: 510-770-9727 <a href="http://www.cyclades.com/%5Cn%5Cn">www.cyclades.com/\n\n</a></td>
</tr>
<tr>
<td>all.prompt</td>
<td>This text defines the format of the login prompt. Expansion characters, listed in Appendix C, can be used here.</td>
<td>%h login:</td>
</tr>
<tr>
<td>all.term</td>
<td>This parameter defines the terminal type assumed when performing rlogin or telnet to other hosts.</td>
<td>vt100</td>
</tr>
<tr>
<td>all.flow</td>
<td>This sets the flow control to hardware, software, or none.</td>
<td>hard</td>
</tr>
<tr>
<td>all.socket_port</td>
<td>This parameter defines the port(s) to be used by the protocols telnet and socket_client. If not configured, a default value of 23 is used.</td>
<td>23</td>
</tr>
<tr>
<td>all.userauto</td>
<td>Username used when connected to a Unix server from the user’s serial terminal.</td>
<td></td>
</tr>
<tr>
<td>s1.tty</td>
<td>The device name for the port is set to the value given in this parameter. If a device name is not provided for a port, it will not function.</td>
<td>ttyS1</td>
</tr>
<tr>
<td>s16.tty</td>
<td>See the s1.tty entry in this table.</td>
<td>ttyS16</td>
</tr>
</tbody>
</table>

**FIGURE 6.10 TERMINAL SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONTINUED)**

Execute the command `signal_ras hup` to activate the changes. At this point, the configuration should be tested. A step-by-step check list follows.

1. Since authentication was set to none, the Cyclades-TS will not authenticate the user. However, the Linux Server receiving the connection will. Create a new user on the server called test and provide him with the password test.
2. From the console, ping 200.200.200.3 to make sure the server is reachable.
3. Make sure that the physical connection between the Cyclades-TS and the terminals is correct. A cross cable (not the modem cable provided with the product) should be used. Please see the hardware specifications appendix for pin-out diagrams.
4. The Cyclades-TS has been set for communication at 9600 bps, 8N1. The terminals must also be configured with the same parameters.
5. From a terminal connected to the Cyclades-TS, try to log in to the server using the username and password configured in item one.

Now continue on to step four later in this chapter.

STEP THREE - REMOTE ACCESS SERVER
The remote access server profile allows a modem user to access the LAN. Radius authentication is used in this example and ppp is chosen as the protocol.

![WARNING! Remote Access Server functionality was only added to provide a secure and effective means of out-of-band access to servers attached to our product. Use exclusively as a Remote Access Server is not advised.]
The fifth configuration file (the first four were described in step two) is specific to the Cyclades-TS and a sample file with comments is supplied in the Linux file system. It is called /etc/portslave/pslave.conf. A listing of pslave.conf with all possible parameters, as well as sample files used to create the three applications in this chapter, is provided in Appendix C. There are three basic types of parameters: conf.* parameters are global or apply to the Ethernet interface; all.* parameters are used to set default parameters for all ports, and s#.* parameters change the default port parameters for individual ports. An all.* parameter can be overridden by a s#. parameter appearing later in the pslave.conf file (or vice-versa). A brief description of each parameter used for the remote access server profile is given in Figures 6.12-6.13.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf.eth_ip</td>
<td>The IP address of the Ethernet interface. This parameter, along with the next two, is used by the cy_ras program to OVERWRITE the file /etc/network/ifcfg_eth0 as soon as the command &quot;signal_ras hup&quot; is executed. The file /etc/network/ifcfg_eth0 should not be edited by the user unless the cy_ras application is not going to be used.</td>
<td>200.200.200.1</td>
</tr>
<tr>
<td>conf.eth_mask</td>
<td>The mask for the Ethernet network.</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>conf.eth_mtu</td>
<td>The Maximum Transmission Unit size, which determines whether or not packets should be broken up.</td>
<td>1500</td>
</tr>
<tr>
<td>conf.lockdir</td>
<td>The lock directory, which is /var/lock for the Cyclades-TS. It should not be changed unless the user decides to customize the operating system.</td>
<td>/var/lock</td>
</tr>
<tr>
<td>conf.pppd</td>
<td>Location of the ppp daemon with Radius/TacacsPlus.</td>
<td>/usr/local/sbin/ pppd</td>
</tr>
<tr>
<td>conf.facility</td>
<td>This value (0-7) is the Local facility sent to the syslog. The file /etc/syslog-ng/syslog-ng.conf contains a mapping between the facility number and the action (see more in Appendix G).</td>
<td>7</td>
</tr>
</tbody>
</table>

*FIGURE 6.12 REMOTE ACCESS SERVER PSLAVE.CONF GLOBAL PARAMETERS*
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.speed</td>
<td>The speed for all ports.  This value (as for any &quot;all.&quot; parameters) can later be overridden for individual ports using the s&lt;port number&gt;.speed parameter.</td>
<td>57600</td>
</tr>
<tr>
<td>all.datasize</td>
<td>The data size for all ports.</td>
<td>8</td>
</tr>
<tr>
<td>all.stopbits</td>
<td>The number of stop bits for all ports</td>
<td>1</td>
</tr>
<tr>
<td>all.parity</td>
<td>The parity for all ports.</td>
<td>none</td>
</tr>
<tr>
<td>all.authtype</td>
<td>There are several authentication type options: local (authentication is performed using the /etc/passwd file), radius (authentication is performed using a Radius authentication server), TacacsPlus (authentication is performed using a TacacsPlus authentication server), none, local/radius (authentication is performed locally first, switching to Radius if unsuccessful), radius/local (the opposite of the previous option), RadiusDownLocal (local authentication is tried only when the Radius server is down), local/TacacsPlus (authentication is performed locally first, switching to TacacsPlus if unsuccessful), TacacsPlus/local (the opposite of the previous option), TacacsPlusDownLocal (local authentication is tried only when the TacacsPlus server is down). Note that this parameter controls the authentication required by the Cyclades-TS. The authentication required by the device to which the user is connecting is controlled separately.</td>
<td>radius</td>
</tr>
<tr>
<td>all.authhost1</td>
<td>This address indicates the location of the Radius/TacacsPlus authentication server and is only necessary if this option is chosen in the previous parameter. A second Radius/TacacsPlus authentication server can be configured with the parameter all.authhost2.</td>
<td>200.200.200.2</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value for This Example</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>all.accthost1</td>
<td>This address indicates the location of the Radius/TacacsPlus accounting server, which can 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 will not be performed. If the same server is used for authentication and accounting, both parameters must be filled with the same address. A second Radius/TacacsPlus accounting server can be configured with the parameter all.accthost2.</td>
<td>200.200.200.2</td>
</tr>
<tr>
<td>all.radtimeout</td>
<td>This is the timeout (in seconds) for a radius/TacacsPlus authentication query. The first server (authhost1) is tried &quot;radretries&quot; times, and then the second (if configured) is contacted &quot;radretries&quot; times. If the second also fails to respond, Radius/TacacsPlus authentication fails.</td>
<td>5</td>
</tr>
<tr>
<td>all.radretries</td>
<td>Defines the number of times each Radius/TacacsPlus server is tried before another is contacted. The default, if not configured, is 5.</td>
<td>5</td>
</tr>
<tr>
<td>all.secret</td>
<td>This is the shared secret necessary for communication between the Cyclades-TS and the Radius/TacacsPlus servers.</td>
<td>cocomero</td>
</tr>
<tr>
<td>all.protocol</td>
<td>For the remote access server profile, the available protocols are PPP, SLIP and CSLIP.</td>
<td>ppp</td>
</tr>
<tr>
<td>all.ipno</td>
<td>The IP address to be assigned to the dial-in users. The &quot;+&quot; indicates that the first port should be addressed as 192.168.1.101 and the following ports should have consecutive values.</td>
<td>200.200.200.11+</td>
</tr>
<tr>
<td>all.netmask</td>
<td>The netmask corresponding to the IP number provided in the previous parameter.</td>
<td>255.255.255.255</td>
</tr>
<tr>
<td>all.mtu</td>
<td>The maximum transmission unit (MTU) that can be transmitted in a PPP packet.</td>
<td>1500</td>
</tr>
<tr>
<td>all.mru</td>
<td>The maximum reception unit (MRU) that can be received in a PPP packet.</td>
<td>1500</td>
</tr>
</tbody>
</table>

*FIGURE 6.13  REMOTE ACCESS SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONTINUED)*
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.initchat</td>
<td>Modem initialization string.</td>
<td><code>TIMEOUT 10 &quot;\d\dATZ \OK\n-ATZ-OK\n &quot; &quot; \&quot;ATMO OK\n\n &quot; \nTIMEOUT 3600 RING &quot; \nSTATUS Incoming %p:I.HANDSHAKE &quot; ATA \nTIMEOUT 60 CONNECT@ &quot; \nSTATUS Connected %p:I.HANDSHAKE</code></td>
</tr>
<tr>
<td>all.flow</td>
<td>This sets the flow control to hardware, software, or none.</td>
<td><code>hard</code></td>
</tr>
<tr>
<td>all.autoppp</td>
<td>PPP options to auto-detect a ppp session.</td>
<td><code>%i:%j novj \ proxyarp modem asyncmap 000A0000 \ noipx noccp login auth require-pap refuse-chap \ mtu %t mru %t \ cb-script /etc/portslave/cb_script \ plugin /usr/lib/libpsr.so</code></td>
</tr>
<tr>
<td>all.pppopt</td>
<td>PPP options when user has already been authenticated.</td>
<td><code>%i:%j novj \ proxyarp modem asyncmap 000A0000 \ noipx noccp mtu %t mru %t netmask %m \ idle %I maxconnect %T \ plugin /usr/lib/libpsr.so</code></td>
</tr>
</tbody>
</table>
FIGURE 6.13 REMOTE ACCESS SERVER PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONTINUED)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1.tty</td>
<td>The device name for the port is set to the value given in this parameter. If a device name is not provided for a port, it will not function.</td>
<td>ttyS1</td>
</tr>
<tr>
<td>s32.tty</td>
<td>See the s1.tty entry in this table.</td>
<td>ttyS32</td>
</tr>
</tbody>
</table>

Execute the command `signal_ras hup` to activate the changes. At this point, the configuration should be tested. A step-by-step check list follows.

1. Since Radius authentication was chosen, create a new user on the Radius authentication server called test and provide him with the password test.

2. From the console, ping 200.200.200.2 to make sure the Radius authentication server is reachable.

3. Make sure that the physical connection between the Cyclades-TS and the modems is correct. The modem cable provided with the product should be used. Please see the hardware specifications appendix for pin-out diagrams.

4. The Cyclades-TS has been set for communication at 57600 bps, 8N1. The modems should be programmed to operate at the same speed on the DTE interface.

5. Try to dial in to the Cyclades-TS from a remote computer using the username and password configured in item one. The computer dialing in must be configured to receive its IP address from the remote access server (the Cyclades-TS in this case) and to use PAP authentication.

Now continue on to step four.
STEP FOUR - FOR ALL PROFILES

TS100 owners, please skip to the special section on the TS100 later in this chapter, then return to this section to continue with step four.

Restart the cy_ras process using its process ID. This can be done by executing the command:

```
signal_ras hup
```

This executes the ps command, searches for the cy_ras process id, then sends the signal HUP to the process, all in one step.

Next, the command `saveconf`, which reads the file `/etc/config_files`, should then be run. The command `saveconf` copies all the files listed in the file `/etc/config_files` from the ramdisk to `/proc/flash/script`. The previous contents of the file `/proc/flash/script` will be lost.

Now the configuration is complete.

| saveconf is equivalent to `tar -czf /proc/flash/script -T /etc/config_files` in standard Linux (saveconf must be used because tar on the TS does not support the z flag). |

| `restoreconf` does the opposite of `saveconf`, copying the contents of the `/proc/flash/script` file to the corresponding files in the ramdisk. The files on the ramdisk are overwritten. `restoreconf` is run automatically each time the Cyclades-TS is booted. |
Information applicable only to the Cyclades-TS100

Since there are two physical interfaces available in the Cyclades-TS100, RS-232 and RS-485, this model requires the configuration of the parameter described in the Figure 6.14.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.media or s1.media</td>
<td>For the TS100 only. <strong>rs232</strong> (RS-232 interface and DB-9 connector), <strong>rs485_half_terminator</strong> (RS-485 interface, half duplex communication with two wires, DB-9 or block connector, the TS100 terminates the network), <strong>rs422</strong> (RS-485 interface, full duplex communication with four wires, DB-9 or block connector, the TS100 terminates the network) or <strong>rs485_half</strong> (RS-485 interface, half duplex communication with two wires, DB-9 or block connector, the TS100 in the middle of the network).</td>
</tr>
</tbody>
</table>

*NOTE:* all.* parameters are used to set default parameters for all ports and s#.* parameters change the default parameters for individual ports. As the TS100 has only one port, either s1.* or all.* can be used, interchangeably.

**FIGURE 6.14 CYCLADES-TS100-MEDIA PARAMETER**

The next step is to update the system with the modified data in the files above. Make sure the file named /etc/config_files contains the names of all files that should be saved to flash.

**Configuring the Cyclades-TS100 for the first time**

The Cyclades-TS100 does not have a dedicated console port. After configuring the serial port, edit the file /etc/inittab and comment the line that designates the console port (add a “#” to the beginning of the line):

```
# ttyS0::respawn:/sbin/getty -p ttyS0 ansi
```

Next, the command **saveconf**, which reads the /etc/config_files file, should be run. The command **saveconf** copies all the files listed in the file /etc/config_files from the ramdisk to /proc/flash/script. The previous contents of the file /proc/flash/script will be lost.
After rebooting the TS100, the initial configuration is complete.

**Clustering**

Clustering has been added to the Cyclades-TS with firmware version 1.3.0 (except for the TS100). It allows the stringing of Terminal Servers so that one master Cyclades-TS can be used to access all Cyclades-TSs on a LAN. The master Cyclades-TS can manage up to 512 serial ports, so

- 1 Master TS1000 + 31 slave TS1000s, or
- 1 Master TS2000 + 15 slave TS2000s, or
- 1 Master TS3000 + 9 slave TS3000s + 1 slave TS2000

can be clustered.

An example with one master TS2000 and two slave TS2000s is shown in Figure 6.15.
FIGURE 6.16 EXAMPLE USING THE CLUSTERING FEATURE.
The Master Cyclades-TS must contain references to the Slave ports. The configuration described earlier for Console Access Servers should be followed with the following exceptions for the Master and Slaves:

Master Configuration:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf.eth_ip</td>
<td>Ethernet Interface IP address.</td>
<td>20.20.20.1</td>
</tr>
<tr>
<td>conf.eth_ip_alias</td>
<td>Secondary IP address for the Ethernet Interface (needed for clustering feature).</td>
<td>209.81.55.110</td>
</tr>
<tr>
<td>conf.eth_mask_alias</td>
<td>Mask for secondary IP address above.</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>all.socket_port</td>
<td>This value applies to both the local ports and ports on slave Cyclades-TSs.</td>
<td>7001+</td>
</tr>
<tr>
<td>all.protocol</td>
<td>Depends on the application.</td>
<td>Socket_ssh or socket_server</td>
</tr>
<tr>
<td>all.authtype</td>
<td>Depends on the application.</td>
<td>Radius or local or none</td>
</tr>
<tr>
<td>s33.tty</td>
<td>This parameter must be created in the master TS file for every slave port.</td>
<td>20.20.20.2:7033</td>
</tr>
<tr>
<td></td>
<td>Its format is IP_of_Slave:[slave_socket_port] for non-master ports.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In this case, the slave_socket_port value is not necessary because s33.socket_port is automatically set to 7033 by all.socket_port above.</td>
<td></td>
</tr>
<tr>
<td>s33.serverfarm</td>
<td>An alias for this port.</td>
<td>Server_on_slave1_serial_s1</td>
</tr>
<tr>
<td>s33.ipno</td>
<td>This parameter must be created in the master TS file for every slave port,</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td></td>
<td>unless configured using all.ipno.</td>
<td></td>
</tr>
<tr>
<td>s34.tty</td>
<td>See s33.tty.</td>
<td>20.20.20.2:7034</td>
</tr>
<tr>
<td>s34.serverfarm</td>
<td>An alias for this port.</td>
<td>Server_on_slave1_serial_s2</td>
</tr>
<tr>
<td>s34.ipno</td>
<td>See s33.ipno.</td>
<td>0.0.0.0</td>
</tr>
</tbody>
</table>

*FIGURE 6.16 MASTER CYCLADES-TS CONFIGURATION (WHERE IT DIFFERS FROM THE STANDARD CAS PROFILE)*
### Table: Parameter Description and Value for This Example

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>s35.tty</td>
<td>See s33.tty.</td>
<td>20.20.2:7035</td>
</tr>
<tr>
<td>s35.serverfarm</td>
<td>An alias for this port.</td>
<td>Server_on_slave1_serial_s3</td>
</tr>
<tr>
<td>s35.ipno</td>
<td>See s33.ipno.</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>etc. for s36-s64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S65.tty</td>
<td>The format of this parameter is IP_of_Slave: [slave_socket_port] for non-master ports. The value 7301 was chosen arbitrarily for this example.</td>
<td>20.20.20.3:7301</td>
</tr>
<tr>
<td>S65.serverfarm</td>
<td>An alias for this port.</td>
<td>Server_on_slave2_serial_s1</td>
</tr>
<tr>
<td>S65.ipno</td>
<td>See s33.ipno.</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>S66.tty</td>
<td>See s65.tty.</td>
<td>20.20.20.3:7302</td>
</tr>
<tr>
<td>S66.serverfarm</td>
<td>An alias for this port.</td>
<td>Server_on_slave2_serial_s2</td>
</tr>
<tr>
<td>S66.ipno</td>
<td>See s33.ipno.</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>S67.tty</td>
<td>See s65.tty.</td>
<td>20.20.20.3:7303</td>
</tr>
<tr>
<td>S67.serverfarm</td>
<td>An alias for this port.</td>
<td>Server_on_slave2_serial_s3</td>
</tr>
<tr>
<td>S67.ipno</td>
<td>See s33.ipno.</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>etc. for s68-s96</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 6.16 MASTER CYCLADES-TS CONFIGURATION (CONT.)**

The Slave Cyclades-TSs do not need to know they are being accessed through the Master Cyclades-TS. Their port numbers, however, must agree with those assigned by the Master.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.protocol</td>
<td>socket_server</td>
</tr>
<tr>
<td>all.authtype</td>
<td>none</td>
</tr>
<tr>
<td>conf.eth_ip</td>
<td>20.20.20.2</td>
</tr>
<tr>
<td>all.socket_port</td>
<td>7033+</td>
</tr>
</tbody>
</table>

**FIGURE 6.17 CYCLADES-TS CONFIGURATION FOR SLAVE 1 (WHERE IT DIFFERS FROM THE STANDARD CAS PROFILE)**
Parameter | Value for This Example
--- | ---
all.protocol | Socket_server
all.authtype | None
conf.eth_ip | 20.20.20.3
all.socket_port | 7301+

**FIGURE 6.18 CYCLADES-TS CONFIGURATION FOR SLAVE 2 (WHERE IT DIFFERS FROM THE STANDARD CAS PROFILE)**

To access ports from the remote management workstation, use telnet with the secondary IP address:

- Telnet 209.81.55.110 7001 to access the first port of the Master Cyclades-TS
- Telnet 209.81.55.110 7033 to access the first port of Slave 1
- Telnet 209.81.55.110 7065 to access the first port of Slave 2

Note that socket port 7065 is being used in the last example to access port 7301 in Slave 2.

ssh can also be used from the remote management workstation:

- ssh -l <username>:Server_on_slave2_serial_s3 209.81.55.110 to access the third port of Slave 2
- ssh -l <username>:7069 209.81.55.110 to access the fifth port of Slave 2

**Centralized Management - Include File**

The Cyclades-TS allows centralized management through the use of a master pslave.conf file. Administrator’s should consider this approach to configure multiple Cyclades-TSSs. Using this feature, each unit has a simplified pslave.conf file where a master include file is cited. This common configuration file contains information for all units, properly separated in separate sections, and would be stored on one central server. This file, in our example shown in figure 6.19, is /etc/portslave/TScommon.conf. It must be downloaded to each Cyclades-TS.
Cyclades-TS

Unit 1
IP address: 10.0.0.1/8

Unit 2
IP address: 10.0.0.2/8

Unit 3
IP address: 10.0.0.3/8

Server where master configuration file is stored
/etc/portslave/TScommon.conf

FIGURE 6.19 EXAMPLE OF CENTRALIZED MANAGEMENT

The abbreviated pslave.conf and /etc/hostname files in each unit, for the example are:

unit 1:

<table>
<thead>
<tr>
<th>Unit</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit 1</td>
<td>10.0.0.1/8</td>
</tr>
<tr>
<td>Unit 2</td>
<td>10.0.0.2/8</td>
</tr>
<tr>
<td>Unit 3</td>
<td>10.0.0.3/8</td>
</tr>
</tbody>
</table>

FIGURE 6.20 /ETC/HOSTNAME FILE IN UNIT 1

<table>
<thead>
<tr>
<th>Conf. Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf.eth_ip</td>
<td>10.0.0.1</td>
</tr>
<tr>
<td>conf.eth_mask</td>
<td>255.0.0.0</td>
</tr>
<tr>
<td>conf.include</td>
<td>/etc/portslave/TScommon.conf</td>
</tr>
</tbody>
</table>

FIGURE 6.21 PSLAVE.CONF FILE IN UNIT 1
The common include file for the example is:
When this file is included, unit1 would read only the information between “conf.host_config  unit1” and conf.host_config  unit2”. Unit2 would use only the information between “conf.host_config  unit2” and conf.host_config  unit3” and unit3 would use information after “conf.host_config  unit3” and before conf.host_config .end.

The following steps should be followed to use centralized configuration:

1. Create and save the /etc/portslave/pslave.conf and /etc/hostname files in each Cyclades-TS
2. Execute the command signal_ras hup on each unit.
3. Create and save the common configuration file on the server, then download it (probably using scp) to each unit. Make sure to put it in the directory set in the pslave.conf file (/etc/portslave in the example).
4. Execute the command signal_ras hup on each unit again.
5. Test each unit. If everything works, add the line /etc/portslave/TScommon.conf to the /etc/config_files file. Save the file and close it. Next, execute the saveconf command.

NOTE: The included file /etc/portslave/TScommon.conf cannot contain another include file (i.e. the parameter conf.include must not be defined).
CHAPTER 7 UPGRADES AND TROUBLESHOOTING

Upgrades

All 6 files added by Cyclades to the standard Linux files are in the /proc/flash directory. They are:
- boot_ori - original boot code
- boot_alt - alternate boot code
- syslog - event logs (not used by Linux)
- config - configuration parameters, only the boot parameters are used by the boot code
- zImage - Linux kernel image
- script - file where all Cyclades-TS configuration information is stored

To upgrade the Cyclades-TS, proceed as follows:

A) Log in to the TS as root (provide the root password if requested)

B) Go to the /proc/flash directory using the following command:
```bash
cd /proc/flash
```

C) Ftp to the host where the new firmware is located, log in using your username and password, go to the directory where the firmware is located, select binary transfer and "get" the firmware file. NOTE: the destination file name in the /proc/flash directory must be zImage. Example (hostname = server; directory = /tftpboot; username = admin; password = adminpw; firmware filename on that server = zImage.132):
```
ftp
> open server
> user admin
> Password: adminpw
> cd /tftpboot
> bin
> get zImage.132 zImage
> quit
```
NOTE: Due to space limitations, the new zImage file may not be downloaded with a different name, then renamed. The TS searches for a file named zImage when booting and there is no room in flash for two zImage files.

To make sure the downloaded file is not corrupted or that the zImage saved in flash is OK, run:

```
md5sum -b /proc/flash/zImage
```

Now check with the information present in the text file saved in the Cyclades site (e.g. zImage.132.md5sum). If the numbers match the downloaded file is not corrupted.

D) Issue the command reboot

```
reboot
```

E) After rebooting, the new Linux kernel will take over. This can be confirmed by typing `cat /proc/version` to see the Linux kernel version.

Troubleshooting

If the contents of flash memory are lost after an upgrade, please follow the instructions below to restore your system:

a. Turn the TS OFF, then back ON
b. Using the console, during self test, press <Esc> after the Ethernet test
c. When the Watch Dog Timer prompt appears, press <Enter>
d. Choose the option Network Boot when asked
e. Enter the IP address of the Ethernet interface
f. Enter the IP address of the host where the new zImage file is located
g. Enter the file name of the zImage file on the host
h. Select the TFTP option instead of BOOTP (the host must be running TFTP and the new zImage file must be located in the proper directory. e.g. /tftpboot for Linux).
i. Accept the default MAC address by pressing <Enter>
j. The TS should begin to boot off the network and the new image will be downloaded and begin running in RAM. At this point, follow the upgrade steps above (login, cd /proc/flash, ftp, and so forth) to save the new zImage file into flash again.

NOTE: possible causes for the loss of flash memory: downloaded wrong zImage file, downloaded as ASCII instead of binary; problems with flash memory.

If the Cyclades-TS booted properly, the interfaces can be verified using ifconfig and ping. If ping does not work, check the routing table using the command route. Of course, all this should be tried after checking that the cables are connected correctly.

As mentioned in Chapter 6, the file /etc/config_files contains a list of files acted upon by saveconf and restoreconf. If a file is missing, it will not be loaded onto the ramdisk on boot. The following table lists files that should be included in the /etc/config_files file and which programs use each.

<table>
<thead>
<tr>
<th>File</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/securetty</td>
<td>telnet, login, su</td>
</tr>
<tr>
<td>/etc/issue</td>
<td>getty</td>
</tr>
<tr>
<td>/etc/getty_ttyS0</td>
<td>login (via console)</td>
</tr>
<tr>
<td>/etc/hostname</td>
<td>tcp</td>
</tr>
<tr>
<td>/etc/hosts</td>
<td>tcp</td>
</tr>
<tr>
<td>/etc/host.conf</td>
<td>tcp</td>
</tr>
<tr>
<td>/etc/nsswitch.conf</td>
<td>dns</td>
</tr>
<tr>
<td>/etc/resolv.conf</td>
<td>dns</td>
</tr>
<tr>
<td>/etc/config_files</td>
<td>saveconf</td>
</tr>
<tr>
<td>/etc/passwd</td>
<td>login, passwd, adduser...</td>
</tr>
<tr>
<td>/etc/group</td>
<td>login, passwd, adduser...</td>
</tr>
<tr>
<td>/etc/ssh/ssh_host_key.pub</td>
<td>sshd</td>
</tr>
<tr>
<td>/etc/ssh/sshd_config</td>
<td>sshd</td>
</tr>
<tr>
<td>/etc/ssh/ssh_config</td>
<td>sshd client</td>
</tr>
<tr>
<td>/etc/ssh/ssh_host_key</td>
<td>sshd (ssh1)</td>
</tr>
<tr>
<td>/etc/ssh/ssh_host_key.pub</td>
<td>sshd (ssh1)</td>
</tr>
</tbody>
</table>
### Hardware Test

A hardware test called `tstest` is included with the Cyclades-TS firmware. It is a menu-driven program, run by typing `tstest` at the command prompt, and the various options are described below.

Note: The Cyclades-TS should not be tested while in use. The user should inactivate all processes that may use the serial ports. They are `inetd`, `sshd`, `cy_ras`, and `cy_buffering`. The user should follow the steps below:

<table>
<thead>
<tr>
<th>File</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/ssh/ssh_host_dsa_key</code></td>
<td><code>sshd</code> (ssh2)</td>
</tr>
<tr>
<td><code>/etc/ssh/ssh_host_dsa_key.pub</code></td>
<td><code>sshd</code> (ssh2)</td>
</tr>
<tr>
<td><code>/etc/snmp/snmpd.conf</code></td>
<td><code>snmpd</code></td>
</tr>
<tr>
<td><code>/etc/portslave/pslave.conf</code></td>
<td><code>cy_ras</code>, <code>portslave</code>, TS configuration information</td>
</tr>
<tr>
<td><code>/etc/network/ifcfg_eth0</code></td>
<td><code>ifconfig eth0, cy_ras, rc.sysinit</code></td>
</tr>
<tr>
<td><code>/etc/network/ifcfg*</code></td>
<td><code>ifconfig</code>, <code>cy_ras, rc.sysinit</code></td>
</tr>
<tr>
<td><code>/etc/network/ifcfg_lo</code></td>
<td><code>ifconfig lo, cy_ras, rc.sysinit</code></td>
</tr>
<tr>
<td><code>/var/run/radsession.id</code></td>
<td><code>radinit</code>, radius authentication process</td>
</tr>
<tr>
<td><code>/home</code></td>
<td><code>adduser, passwd</code></td>
</tr>
<tr>
<td><code>/etc/network/st_routes</code></td>
<td><code>ifconfig</code>, <code>cy_ras, rc.sysinit</code></td>
</tr>
<tr>
<td><code>/etc/syslog-ng/syslog-ng.conf</code></td>
<td><code>syslog-ng</code></td>
</tr>
</tbody>
</table>

If any of the files listed in `/etc/config_files` is modified, the Cyclades-TS administrator must execute the command `saveconf` before rebooting the Cyclades-TS or the changes will be lost. If a file is created (or a file name altered), its name must be added to this file before executing `saveconf` and reboot.

Cyclades Technical Support is always ready to help with any configuration problems. Before calling, execute the command

```bash
cat /proc/version
```

and note the Linux version and Cyclades-TS version written to the screen. This will speed resolution of most problems.
step 1) signal_ras stop.

step 2) Perform all hardware tests needed.

step 3) signal_ras start.

**Port Test**

Either a cross cable or a loop-back connector is necessary for this test. Their pinout diagrams are supplied in appendix B. Connect the loop-back connector to the modem cable and then connect the modem cable to the port to be tested (or connect a cross cable between two ports to be tested). In the case of the TS100, connect the DB-25 loop-back connector to the console cable using a DB-9 - DB-25 convertor. When ts test senses the presence of the cable or connector, the test will be run automatically and the result shown on the screen.

Each line of data corresponds to a port in test. The last 4 columns (DATA, CTS, DCD, and DSR) indicate errors. The values in these columns should be zero. The figure below is an example of the output screen.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Sent</th>
<th>Received</th>
<th>Passes</th>
<th>Data</th>
<th>CTS</th>
<th>DCD</th>
<th>DSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

When this test is run with a cable or connector without the DSR signal (see the pinout diagram for the cable or connector being used), errors will appear in the DSR column. This does not indicate a problem with the port. In the example above, ts test perceived that a loop-back connector was attached to port 2 and that a cross cable was used to connect ports 4 and 5.

**Port Conversation**

This test sends and receives data on the selected port. One way to run this test is to place a loop-back connector on the port to be tested and begin. Enter the number of the port and a baud rate (9600 is a typical value). Type some letters, and if the letters appear on the screen, the port is working. If the letters do not appear
on the screen (which also occurs if the loop-back connector is removed), the port is not functioning correctly.

A second method that can be used to test the port is to connect it to a modem with a straight cable. Begin the test and type “at”. The modem should respond with “OK”, which will appear on the screen. Other commands can be sent to the modem or to any other serial device.

**Test Signals Manually**

This test confirms that signals are being sent and received on the selected port. Neither the loop-back connector nor the cross cable are necessary. Enter the number of the port to be tested and begin the test.

<table>
<thead>
<tr>
<th>State</th>
<th>DTR</th>
<th>DCD</th>
<th>DSR</th>
<th>RTS</th>
<th>CTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>↓</td>
<td></td>
<td></td>
<td>↓</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

First, type Ctrl-D to see the X in the DTR column move position, then type Ctrl-R to see the X in the RTS column change position. If each of the Xs moves in response to its command, the signals are being sent.

Another method to test the signals is to use a loop-back connector. Enter the number of the port with the loop-back connector and start the test. In this case, when Ctrl-D is typed, the Xs in the first three columns will move as shown below.

<table>
<thead>
<tr>
<th>State</th>
<th>DTR</th>
<th>DCD</th>
<th>DSR</th>
<th>RTS</th>
<th>CTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>OFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This is because the test is receiving the DTR signal sent through the DCD and DSR pins. When Ctrl-R is typed, the Xs in the RTS and CTS columns should move together. If the Xs change position as described, the signals are being sent and received correctly.

**Single User Mode**

The Cyclades-TS has a single user mode used when:
- The name or password of the user with root privileges is lost or forgotten,
- After an upgrade or downgrade which leaves the Cyclades-TS unstable,
- After a configuration change which leaves the Cyclades-TS inoperative or unstable.

Type the word “single” (with a blank space before the word) during boot using a console connection. This cannot be done using a telnet or other remote connection.

The initial output of the boot process is shown below.

```
Entry Point = 0x00002120
loaded at: 00002120 0000D370
relocated to: 00300020 0030B270
board data at: 003052C8 0030537C
relocated to: 002FF120 002FF1D4
zimage at: 00008100 0006827E
relocated to: 00DB7000 00E1717E
initrd at: 0006827E 0024F814
relocated to: 00E18000 00FF596
avail ram: 0030B270 00E18000
Linux/PPC load: root=/dev/ram
```

After printing “Linux/PPC load: root=/dev/ram”, the Cyclades-TS waits approximately 10 seconds for user input. This is where the user should type “<sp>single” (spacebar, then the word “single”). When the boot process is complete, the Linux prompt will appear on the console:

```
[root@(none) /]#```
If the password or username was forgotten, execute the following commands:

```
passwd
saveconf
reboot
```

For configuration problems, the user has two options:

1. Edit the file(s) causing the problem with vi, then execute the commands:

```
saveconf
reboot
```

2. Reset the configuration by executing the commands:

```
echo 0 > /proc/flash/script
reboot
```

If the problem is due to an upgrade/downgrade, a second downgrade/upgrade will be necessary to reverse the process. First, the network must be initialized in order to reach a ftp server. Execute the following script, replacing the parameters with values appropriate for your system. If your ftp server is on the same network as the TS, the gw and mask parameters are optional.

```
config_eth0 ip 200.200.200.1 mask 255.255.255.0 gw 200.200.200.5
```

At this point, the DNS configuration (in the file /etc/resolv.conf) should be checked. Then, download the kernel image using the ftp command.

**Troubleshooting the Web Configuration Manager**

1. **What to do when the initial web page does not appear.**

Try pinging, telnetting or tracerouting to the Cyclades-TS to make sure it is reachable. If not, the problem is probably in the network or network configuration. Are the interfaces up? Are the IP addresses correct? Are filters configured which block the packets?
If the Cyclades-TS is reachable, see if the /bin/webs process is running by executing the command `ps`. If it is not, type `/bin/webs &` to start it. If the /bin/webs process is not being initialized during boot, change the file `/etc/inittab`.

2. How to restore the default configuration of the Web Configuration Manager

This would be required only when the root password was lost or the configuration file `/etc/websum.conf` was damaged.

From a console or telnet session, edit the file `/etc/config_files`. Find the reference to `/etc/websum.conf` and delete it. Save the modified `/etc/config_files` file. Execute the command `saveconf`. Reboot the system. Enter into the Web Configuration Manager with the default username and password (`root/tslinux`). Edit the file `/etc/config_files` and insert the reference to `/etc/websum.conf`.

Recover the access to the Cyclades-TS100 console port

There is no dedicated console port available in the Cyclades-TS100. As factory default the serial port is set to work as a console port to allow initial product configuration. After that, changes can still be made through the Ethernet port and a Telnet command. If for some reason this access is lost (usually misconfiguration), the product can only be configured if the steps bellow are followed.

1. Power the Cyclades-TS100 off.
2. Connect the Cyclades-TS100 to a terminal configured to work at 9600 bps, with 8 bits, no parity and 1 stop bit.
3. Press and hold the reset button and power on the Cyclades-TS100. Release the reset button when the self test starts on the terminal’s screen.

The Cyclades-TS100 will be now in single user mode, the serial port will work as a console port and the product can be reconfigured. Notice that no previous configuration is lost. After finishing, **save the configuration** (saveconf), power the Cyclades-TS100 off, and reconnect the original device to the serial port. Using a different speed for the Serial Console.
Using a different speed for the serial console
The serial console is originally configured to work at 9600 bps. If the customer wants to change that, it is necessary to run bootconf. The user will be presented with the screen:

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 Maximun Interrupt Events [0]

Type <Enter> for all fields but the Console Speed. When presented the following line:

Do you confirm these changes in flash ( (Y)es, (N)o (Q)uit ) [N] :

Enter Y and the changes will be saved in flash. Reboot the unit to have the changes effective and use the console at the new speed.
APPENDIX A  INFORMATION FOR USERS NOT FAMILIAR WITH LINUX

Users and Passwords
A username and password are necessary to log in to the Cyclades-TS. The user “root” is predefined, with a password tslinux. A password should be configured as soon as possible to avoid unauthorized access. Type the command:

    passwd

to create a password for the root user.

To create a regular user (without root privileges), use the commands:

    adduser user_name
    passwd user_name

NOTE: If you do not use a combination of upper and lower case letters and numbers, and between 5 and 8 characters for the password, you will get a warning, but it will still be accepted.

To log out, type “logout” at the command prompt.

Linux File Structure
The Linux file system is organized hierarchically, with the base (or root) directory represented by the symbol “/”. All folders and files are nested within each other below this base directory. The directories located just below the base directory are:
<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home</td>
<td>Contains the work directories of system users.</td>
</tr>
<tr>
<td>/bin</td>
<td>Contains applications and utilities used during system initialization.</td>
</tr>
<tr>
<td>/dev</td>
<td>Contains files for devices and ports.</td>
</tr>
<tr>
<td>/etc</td>
<td>Contains configuration files specific to the operating system.</td>
</tr>
<tr>
<td>/lib</td>
<td>Contains shared libraries.</td>
</tr>
<tr>
<td>/proc</td>
<td>Contains process information</td>
</tr>
<tr>
<td>/mnt</td>
<td>Contains information about mounted disks.</td>
</tr>
<tr>
<td>/opt</td>
<td>Location where packages not supplied with the operating system are stored.</td>
</tr>
<tr>
<td>/tmp</td>
<td>Location where temporary files are stored.</td>
</tr>
<tr>
<td>/usr</td>
<td>Contains most of the operating system files.</td>
</tr>
<tr>
<td>/var</td>
<td>Contains operating system data files.</td>
</tr>
</tbody>
</table>

### Basic File Manipulation Commands

The basic file manipulation commands allow the user to copy, delete and move files and create and delete directories.

**cp** `file_name destination`  
a) `cp text.txt /tmp`  
b) `cp /chap/robo.php ./excess.php`

Copies the file indicated by `file_name` to the path indicated by `destination`.  
a) copies the file text.txt in the current directory to the tmp directory.  
b) 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`.

**mkdir** `directory_name`  
a) `mkdir spot`  
b) `mkdir /tmp/snuggles`

Creates a directory named `directory_name`.  
a) creates the directory spot in the current directory.  
b) creates the directory snuggles in the directory tmp.

**rmdir** `directory_name`  
Removes the directory indicated by `directory_name`.
Other commands allow the user to change directories and see the contents of a directory.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pwd</code></td>
<td>Supplies the name of the current directory. While logged in, the user is always &quot;in&quot; a directory. The default initial directory is the user's home directory, /home/&lt;username&gt;</td>
</tr>
<tr>
<td><code>ls [options] directory_name</code></td>
<td>Lists the files and directories within <code>directory_name</code>. Some useful options are <code>-l</code> for more detailed output and <code>-a</code> which shows hidden system files.</td>
</tr>
<tr>
<td><code>cd directory_name</code></td>
<td>Changes the directory to the one specified</td>
</tr>
<tr>
<td><code>cat file_name</code></td>
<td>Prints the contents of <code>file_name</code> to the screen.</td>
</tr>
</tbody>
</table>

**Shortcuts:**

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>. (a dot)</td>
<td>represents the current directory</td>
</tr>
<tr>
<td>.. (two dots)</td>
<td>represents one directory above the current directory (i.e. one directory closer to the base directory).</td>
</tr>
</tbody>
</table>

**The vi Editor**

To edit a file using the vi editor, type `vi file_name`

vi is a three-state line editor: it has a command mode, a line mode and an editing mode. If in doubt as to which mode you are in, press the <ESC> key which will bring you to the command mode.

<table>
<thead>
<tr>
<th>Mode</th>
<th>What is done there</th>
<th>How to Get There</th>
</tr>
</thead>
<tbody>
<tr>
<td>command mode</td>
<td>navigation within the open file</td>
<td>Press the &lt;ESC&gt; key.</td>
</tr>
<tr>
<td>editing mode</td>
<td>text editing</td>
<td>See list of editing commands below.</td>
</tr>
<tr>
<td>line mode</td>
<td>file saving, opening, etc. exiting from vi</td>
<td>From the command mode, type &quot;:.&quot; (the colon).</td>
</tr>
</tbody>
</table>
Entering the program, the user is automatically in the command mode. To navigate to the part of the file to be edited, use the following keys:

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>moves the cursor to the left (left arrow)</td>
</tr>
<tr>
<td>j</td>
<td>moves the cursor to the next line (down arrow)</td>
</tr>
<tr>
<td>k</td>
<td>moves the cursor to the previous line (up arrow)</td>
</tr>
<tr>
<td>l</td>
<td>moves the cursor to the right (right arrow)</td>
</tr>
</tbody>
</table>

Having arrived at the location where text should be changed, use these commands to modify the text (note commands “i” and “o” will move you into the editing mode and everything typed will be taken literally until you press the <ESC> key to return to the command mode)

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>insert text before the cursor position (everything to the right of the cursor is shifted right)</td>
</tr>
<tr>
<td>o</td>
<td>create a new line below the current line and insert text (all lines are shifted down)</td>
</tr>
<tr>
<td>dd</td>
<td>remove the entire current line</td>
</tr>
<tr>
<td>u</td>
<td>undo the last modification</td>
</tr>
<tr>
<td>x</td>
<td>delete the letter at the cursor position</td>
</tr>
</tbody>
</table>

Now that the file has been modified, enter the line mode (by typing “:” from the command mode) and use one of the following commands:

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>save the file (w is for write)</td>
</tr>
<tr>
<td>wq</td>
<td>save and close the file (q is for quit)</td>
</tr>
<tr>
<td>q!</td>
<td>close the file without saving</td>
</tr>
<tr>
<td>w file</td>
<td>save the file with the name file</td>
</tr>
<tr>
<td>e file</td>
<td>opens the file named file</td>
</tr>
</tbody>
</table>
The Routing Table

The Cyclades-TS has a static routing table that can be seen using the commands

```
route
```

or

```
netstat -rn
```

The file `/etc/network/st_routes` shown in Figure 6.5 is the Cyclades-TS’s method for configuring static routes. Routes should be added to the file (which is a script run when the Cyclades-TS is initialized) or at the prompt (for temporary routes) using the following syntax:

```
```

| [add|del]   | one of these tags must be present -- routes can be either added or deleted. |
| [-net|-host]| -net is for routes to a network and -host is for routes to a single host. |
| target     | `target` is the IP address of the destination host or network              |
| netmask    | the tag netmask and a mask are necessary only when subnetting is used.  Otherwise, a mask appropriate to the `target` is assumed. `nt_msk` must be specified in dot notation. |
| nt_msk     |                                                                           |
| gw gt_way  | specifies a gateway, when applicable. `gt_way` is the IP address or hostname of the gateway. |
| interf     | the interface to use for this route. Must be specified if a gateway is not. When a gateway is specified, the operating system determines which interface is to be used. |

ssh - The Secure Shell Session

ssh is a command interface and protocol often used by network administrators to connect securely to a remote computer. ssh replaces its non-secure counterpart rsh and rlogin. There are two versions of the protocol, ssh and ssh2. The Cyclades-TS offers both.
The command to start an ssh client session from a Unix workstation is:

```
ssh -t <user>@<hostname>
```

where

```
<user> = <username>:ttySnn or
          <username>:socket_port or
          <username>:ip_addr or
          <username>:serverfarm
```

Note: “serverfarm” is a physical port alias. It can be configured in the file pslave.conf.

An example:

```
username: cyclades
TS1000 IP address: 192.168.160.1
host name: ts1000
servername for port 1: file_server
```

ttyS1 addressed by IP 10.0.0.1 or socket port 7001. The various ways to access the server connected to the port are:

```
ssh -t cyclades:ttyS1@ts1000
ssh -t cyclades:7001@ts1000
ssh -t cyclades:10.0.0.1@ts1000
ssh -t cyclades:file_server@ts1000
```

Note that either `-l` or `@` are used, but not both. For openssh version 3.1p1 or later (Cyclades-TS V_1.3.2 or later), ssh2 is the default. In that case, the `-l` flag is used for ssh1.

```
ssh -t cyclades:7001@ts1000 (openssh earlier than 3.1p1 - Cyclades-TS V_1.3.1 and earlier -> ssh1 will be used)
ssh -t -2 cyclades:7001@ts1000 (openssh earlier than 3.1p1 - Cyclades-TS V_1.3.1 and earlier -> ssh2 will be used)
```
ssh -t cyclades:7001@ts1000 (openssh 3.1p1 or later - Cyclades-TS V_1.3.2 or later -> ssh2 will be used)
ssh -t -l cyclades:7001@ts1000 (openssh 3.1p1 or later - Cyclades-TS V_1.3.2 or later -> ssh1 will be used)

To log in to a port that does not require authentication, the username is not necessary:
ssh -t -2 :ttyS1@ts1000

Note: In this case, the file sshd_config must be changed in the following way:
PermitRootLogin Yes
PermitEmptyPassword Yes

**Configuring sshd’s client authentication using SSH Protocol version 1**

1. Only RhostsAuthentication yes in sshd_config

   - One of these:
     
     hostname or ipaddress in /etc/hosts.equiv or /etc/ssh/hosts.equiv
     
     hostname or ipaddress and username in ~/.rhosts or ~/.shosts and IgnoreRhosts no in sshd_config

   - Client start-up command: ssh -t <TS_ip or Serial_port_ip> (if the ssh client is running under a session belonging to a username present both in the workstation’s database and the TS’s database)

   - Client start-up command: ssh -t -l <username> <TS_ip or Serial_port_ip> (if the ssh client is running under a session belonging to a username present only in the workstation’s database. In this case, the <username> indicated would have to be a username present in the TS’s database)
Note 1: Some ssh clients do not allow just this type of authentication, for security reasons.
Note 2: To access the serial port, the TS must be configured for local authentication.
Note 3: No root user should be used as username.

2. Only RhostsRSAAuthentication yes in sshd_config

- One of the RhostsAuthentication above settings
- Client machine’s host key ($ETC/ssh_host_key.pub) copied into the TS /tmp/known_hosts file. The client hostname plus the information inside this file must be appended in one single line inside the file /etc/ssh/ssh_known_hosts or ~/.ssh/known_hosts and IgnoreUserKnownHosts no inside sshd_config. The following commands can be used for example:

  echo –n “client_hostname “ >> /etc/ssh/ssh_known_hosts or ~/.ssh/known_hosts

  cat /tmp/known_hosts >> /etc/ssh/ssh_known_hosts or ~/.ssh/known_hosts

- client start-up command: ssh -t <TS_ip or Serial_port_ip>

  Note 1: “client_hostname” should be the DNS name.
  Note 2: To access the serial port, the TS must be configured for local authentication.
  Note 3: No root user should be used as username.

3. Only RSAAuthentication yes in sshd_config

- Removal of TS’s *.equiv, ~/.?hosts, and *known_hosts files
- client identity created by ssh-keygen and its public part (~/.ssh/identity.pub) copied into TS’s ~/.ssh/authorized_keys
4. Only PasswdAuthentication yes in sshd_config

- Removal of TS's *.equiv, ~/.?hosts, *known_hosts, and *authorized_keys files

- client startup command: ssh -t <username> <TS_ip or Serial_port_ip> or ssh -t -l <username:alias> <TS_ip>

**Configuring sshd’s client authentication using SSH Protocol version 2**
1. Only PasswdAuthentication yes in sshd_config DSA Authentication is the default (Make sure the parameter PubkeyAuthentication is enabled)

- Client DSA identity created by ssh-keygen -d and its public part (~/.ssh/id_dsa.pub) copied into TS’s ~/ .ssh/authorized_keys2 file

- Password Authentication is performed if DSA key is not known to the TS.
  client start-up command: ssh -2 -t <TS_ip or Serial_port_ip>

**Notice:**

All files “~/*” or “~/.ssh/*” must be owned by the user and readable only by others. All files created or updated must have their full path and file name inside the file config_files and the command saveconf must be executed before rebooting the TS.

**The Process Table**
The process table shows which processes are running. Type ps -a to see a table similar to that below.
<table>
<thead>
<tr>
<th>PID</th>
<th>Uid</th>
<th>State</th>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>root</td>
<td>S</td>
<td>/sbin/inetd</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>root</td>
<td>S</td>
<td>/sbin/sshd</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>root</td>
<td>S</td>
<td>/sbin/cy_ras</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>root</td>
<td>S</td>
<td>/sbin/cy_wdt_led wdt led</td>
<td></td>
</tr>
<tr>
<td>154</td>
<td>root</td>
<td>R</td>
<td>/ps -a</td>
<td></td>
</tr>
</tbody>
</table>

To restart the cy_ras process use its process ID or execute the command:

```
signal_ras hup
```

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.

### NTP Client Functionality

In order for the Cyclades-TS to work as a NTP (Network Timer Protocol) client, the IP address and either hostname or domain name of the NTP server must be set in the file `/etc/hosts`. The date and time will be updated from the NTP server after rebooting.

### The Crond Utility

To use crond, first create the following two files for every process that it will execute:

1. `crontab` - the file that specifies frequency of execution, name of shell script, etc. should be set using the traditional crontab file format.
2. `script shell` - a script file with the Linux commands to be executed.

Next, create a line in the file `/etc/crontab_files` for each process to be run. Each line must contain the three items:
status (active or inactive) - if this item is not active, the script will not be executed.
user - the process will be run with the privileges of this user, who must be a valid local user.
source - pathname of the crontab file.

When the /etc/crontab_files file contains the following line:
   active root /etc/tst_cron.src
and the /etc/tst_cron.src file contains the following line:
   0-59 * * * * /etc/test_cron.sh
crond will execute the script listed in test_cron.sh with root privileges each minute.

Example files are in the /etc directory.

The next step is to update the system with the modified data in the files above and reboot the Cyclades-TS. Make sure the file named /etc/config_files contains the names of all files that should be saved to flash. Next, the command saveconf, which reads the /etc/config_files file, should then be run.

saveconf copies all the files listed in the file /etc/config_files from the ramdisk to /proc/flash/script. See step 5 in chapter 6 for more details.

The DHCP (Dynamic Host Configuration Protocol) Client
(Note: This feature is only available for firmware versions 1.2.x and above)
DHCP is a protocol that allows network administrators to assign IP addresses automatically to network devices. Without DHCP (or a similar protocol like BOOTP), each device would have to be manually configured. DHCP automatically sends a new IP address to a connected device when it is moved to another location on the network. DHCP uses the concept of a fixed time period during which the assigned IP address is valid for the device it was assigned for. This “lease” time can vary for each device. A short lease time can be used when there are more devices than available IP numbers. For more information, see RFC 2131.
The DHCP client on the Ethernet Interface can be configured in two different ways, depending on the action the Cyclades-TS should take in case the DHCP server does not answer the IP address request:

1. No action is taken and no IP address is assigned to the Ethernet Interface (most common configuration):
   - Set the global parameter `conf.dhcp_client` to 1
   - Comment all other parameters related to the Ethernet Interface (conf.eth_ip, etc.)
   - Add the necessary options to the file `/etc/network/dhcpcd_cmd` (some options are described below)

2. The Cyclades-TS restores the last IP address previously provided in another boot and assigns this IP address to the Ethernet Interface:
   - Set the global parameter `conf.dhcp_client` to 2
   - Comment all other parameters related to the Ethernet Interface (conf.eth_ip, etc.)
   - Add the following lines to the file `/etc/config_files`
     /etc/network/dhcpcd_cmd
     /etc/dhcpcd-eth0.save
   - Add the option “-x” to the factory default content of the file `/etc/network/dhcpcd_cmd`:
     `/sbin/dhcpcd -x -c /sbin/handle_dhcp`
   - Add all other necessary options to the file `/etc/network/dhcpcd_cmd` (some options are described below)

In both cases if the IP address of the Cyclades-TS or the default gateway are changed, the Cyclades-TS will adjust the routing table accordingly.
Two files are related to DHCP:

/bin/handle_dhcp - the script which is run by the DHCP client each time an IP address negotiation takes place.

/etc/network/dhcpcd_cmd - contains a command that activates the DHCP client (used by the cy_ras program).

Its factory contents are:

/sbin/dhcpcd -c /sbin/handle_dhcp

The options available that can be used on this command line are:

-D This option forces dhcpcd to set the domain name of the host to the domain name parameter sent by the DHCP server. The default option is to NOT set the domain name of the host to the domain name parameter sent by the DHCP server.

-H This option forces dhcpcd to set the host name of the host to the hostname parameter sent by the DHCP server. The default option is to NOT set the host name of the host to the hostname parameter sent by the DHCP server.

-R This option prevents dhcpcd from replacing the existing /etc/resolv.conf file.

The user should not modify the -c /sbin/handle_dhcp option.

Data Buffering

Since version 1.3.2 of the Cyclades-TS software, additional ramdisks can be created and used, for example, to buffer data. This removed the previous 700 kbyte restriction for all TS ports. Data buffering files are created in the directory /var/run/DB. Previously, data buffering files were named ttyS<nn>.data (where <nn> is the port number). Now, if the parameter s<nn>.serverfarm is configured for the port <nn>, this name will be used. For example, if the serverfarm is called bunny, the data buffering file will be named bunny.data.
The shell script `/bin/build_DB_ramdisk` creates a 4 Mbyte ramdisk for the TS3000. Use this script as a model to create customized ramdisks for your environment. Any user-created scripts should be listed in the file `/etc/user_scripts` because `rc.sysinit` executes all shell scripts found there. This avoids changing `rc.sysinit` itself.

Data buffering can 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.

The user may want 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 as seen in the earlier chapters, and `syslog-ng.conf` file should be set accordingly (please see Appendix G for syslog-ng configuration file). For file, `all.data_buffering` is the parameter to be dealt with as seen in the early chapters.

### Packet Filtering using ipchains

*(Note: This feature is only available for firmware versions 1.2.x and above)*

The Cyclades-TS uses the Linux utility `ipchains` to filter IP packets entering, leaving and passing through its interfaces. An `ipchains` tutorial is beyond the scope of this manual. For more information on `ipchains`, see the `ipchains` man page (not included with the Cyclades-TS) or the howto: [http://netfilter.filewatcher.org/ipchains/HOWTO.html](http://netfilter.filewatcher.org/ipchains/HOWTO.html).

The syntax of the `ipchains` command is:

```
```

where **command** is one of the following:

- **A** - Add a condition or rule to the end of the chain. Note that the order in which a condition appears in a chain can modify its application and the first rule added to a chain is processed first, etc.
- **D** - Delete a condition from the chain. The condition must match exactly with the command’s arguments to be deleted.
- **R** - Replace a condition in the chain.
I - Insert a condition in a specified location in the chain.
L - List all conditions in the chain.
F - Flush (remove) all conditions in the chain.
N - Create a new chain.
X - Deletes a user-created chain
P - Policy applied for default handling

chain is one of the following:
- input - filters incoming packets
- output - filters outgoing packets
- forward - filters packets which are not created by the Cyclades-TS and are not destined to the Cyclades-TS
- user_created_chain - a previously defined (or in the process of being defined) chain created using the N command described above.

The output chain controls which packets are sent. A packet can be accepted by the input chain, but then rejected by the output chain. Likewise, the forward chain controls which packets will be routed. The input chain controls incoming packet filtering. The packet is either destined for the router or for another computer. In the latter case, the packet is processed by the forward chain. Packets that pass through the forward chain will then be processed by the output chain.

source and destination have the following format:
- [!]address[/mask] [!][:port[:port]]
- ! : reverses the definition, resulting in the opposite.
- address : host or network IP
- port : defines a specific port
- port:port : defines a range of ports
- If a source or destination is not specified then 0.0.0.0/0 is used.

protocol is one of the following:
tcp, udp, icmp, all or a protocol number (see the file /etc/protocols for a list).
**target** is one of the following:
- ACCEPT
- DENY
- the name of another chain

**interface** is:
- eth0 (The Ethernet interface is the only option on the Cyclades-TS.) Lists do not need to be associated to an interface, so this option may be omitted.

To save changes made using the ipchains command, execute fwset. This command will save the filter configuration in the file /etc/network/firewall and then save the file in flash memory.

To delete the changes made (before fwset is executed) execute fwset restore to return to the lists previously saved in /etc/network/firewall. Only the lists previously saved using fwset will then be defined. This command is executed at boot to invoke the last configuration saved.

Another option is to edit the file /etc/network/firewall (or another file) directly, following the syntax defined in the file itself. If the file is edited in this way, the command fwset cannot be used to save and restore the configuration. Use

```
ipchains-save > file_name to save the lists in file_name
updatefiles file_name to save file_name to flash memory
ipchains-restore < file_name to restore the lists to the configuration in file_name
```

**An example of the use of ipchains for a console access server**

Referring to Fig 5.5

If the administrator wishes to restrict access to the consoles connected to the Cyclades-TS to a user on the workstation with IP address 200.200.200.4, a filter can be set up as shown below.

```
ipchains -P input ACCEPT
ipchains -P output ACCEPT
ipchains -P forward ACCEPT
ipchains -A input -p tcp -s ! 200.200.200.4 -d 0.0.0.0/0 23 -j DENY
```
The ts_menu script can be used to avoid typing long telnet or ssh commands. It presents a short menu with the names of the servers connected to the serial ports of the Cyclades-TS. The server is selected by its corresponding number. ts_menu must be executed from a local session: via console, telnet, ssh, dumb terminal connected to a serial port, etc.

Only ports configured for console access (protocols socket_server or socket_ssh) will be presented. To start having familiarity with this application, run ts_menu - h:

```
> ts_menu -h

USAGE: ts_menu options

-p : Display Ethernet Ip and Tcp port
-i : Display local Ip assigned to the serial port
-u <name> : Username to be used in ssh/telnet command
-U : Allows choosing of different usernames for different ports
-h : print this help message
```
Selecting 1 in this example, and the user will access the local serial ports on that Cyclades-TS. In case the user selects 2 through 5, remote serial ports will be accessed. This is used when there is clustering (one Cyclades-TS master box and one or more Cyclades-TS slave boxes).

In case the user selects 1 the possible screen to be displayed would be:

```
Serial Console Server Connection Menu for your Master Terminal Server

1 ttyS1  2 ttyS2  3 s3serverfarm

Type 'q' to quit, 'b' to return to previous menu, a valid option [1-3], or anything else to refresh:
```

Options 1 to 3 in this case are serial ports configured to work as CAS profile. Serial port 3 is presented as an alias name (s3serverfarm). When no name is configured in pslave.conf, ttyS<N> is used instead.

Once selected the serial port, the username and password for that port (in case there is a per user access to the port and -U is passed as parameter) will be presented. Otherwise, the access is granted.
To access remote serial ports, the presentation will follow a similar approach as the one used for local serial ports.

The `ts_menu` script has the following line options:

- **-p**: Displays Ethernet IP Address and TCP port instead of server names

```
Cyclades-TS: Serial Console Server Connection menu
1 209.81.55.79 7001 2 209.81.55.79 7002 3 209.81.55.79 7003
4 209.81.55.79 7004 5 209.81.55.79 7005 6 209.81.55.79 7006
Type 'q' to quit, a valid option [1-6], or anything else to refresh:
```

- **-i**: Displays Local IP assigned to the serial port instead of server names

```
Cyclades-TS: Serial Console Server Connection menu
1 192.168.1.101 2 192.168.1.102 3 192.168.1.103 4 192.168.1.104
5 192.168.1.105 6 192.168.1.106
Type 'q' to quit, a valid option [1-6], or anything else to refresh:
```

- **-u <name>**: Username to be used in ssh/telnet command. The default username is that used to log in to the Cyclades-TS.

- **-h**: lists script options
APPENDIX B  HARDWARE SPECIFICATIONS AND CABELING

General Hardware Specifications
The power requirements, environmental conditions and physical specifications of the Cyclades-TS are listed in the table below.

<table>
<thead>
<tr>
<th>POWER SPECIFICATIONS</th>
<th>TS100</th>
<th>TS400</th>
<th>TS800</th>
<th>TS1000</th>
<th>TS2000</th>
<th>TS3000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Voltage Range</td>
<td>External Universal Input Desktop Power Supply (100-240VAC auto-range input, 5VDC output)</td>
<td>External Universal Input Desktop Power Supply (100-240VAC auto-range input, 5VDC output)</td>
<td>External Universal Input Desktop Power Supply (100-240VAC auto-range input, 5VDC output)</td>
<td>Internal 100-240VAC auto-range (-48VDC option available)</td>
<td>Internal 100-240VAC auto-range (-48VDC option available)</td>
<td>Internal 100-240VAC auto-range</td>
</tr>
<tr>
<td>Input Frequency Range</td>
<td>50/60Hz</td>
<td>50/60Hz</td>
<td>50/60Hz</td>
<td>50/60Hz</td>
<td>50/60Hz</td>
<td>50/60Hz</td>
</tr>
<tr>
<td>Power @120VAC</td>
<td>5 W max</td>
<td>5 W max</td>
<td>6 W max</td>
<td>22 W max</td>
<td>26 W max</td>
<td>11 W max</td>
</tr>
<tr>
<td>Power @220VAC</td>
<td>6 W max</td>
<td>6 W max</td>
<td>8 W max</td>
<td>28 W max</td>
<td>37 W max</td>
<td>17 W max</td>
</tr>
</tbody>
</table>

ENVIRONMENTAL INFORMATION

<table>
<thead>
<tr>
<th>ENVIRONMENTAL INFORMATION</th>
<th>TS100</th>
<th>TS400</th>
<th>TS800</th>
<th>TS1000</th>
<th>TS2000</th>
<th>TS3000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Temperature</td>
<td>40F to 104F (10°C to 40°C)</td>
<td>40F to 104F (10°C to 40°C)</td>
<td>40F to 104F (10°C to 40°C)</td>
<td>40F to 104F (10°C to 40°C)</td>
<td>40F to 104F (10°C to 40°C)</td>
<td>40F to 104F (10°C to 40°C)</td>
</tr>
<tr>
<td>Relative Humidity</td>
<td>10 to 90%, non-condensing</td>
<td>10 to 90%, non-condensing</td>
<td>10 to 90%, non-condensing</td>
<td>10 to 90%, non-condensing</td>
<td>10 to 90%, non-condensing</td>
<td>10 to 90%, non-condensing</td>
</tr>
</tbody>
</table>
### PHYSICAL SPECIFICATIONS

<table>
<thead>
<tr>
<th></th>
<th>TS100</th>
<th>TS400</th>
<th>TS800</th>
<th>TS1000</th>
<th>TS2000</th>
<th>TS3000</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Dimensions</td>
<td>2.76in x 3.35in x 1.18in</td>
<td>8.5in x 4.75in x 1in</td>
<td>8.5in x 4.75in x 1in</td>
<td>17in x 8.5in x 1.75in</td>
<td>17in x 8.5in x 1.75in</td>
<td>17in x 8.5in x 1.75in</td>
</tr>
<tr>
<td>Weight</td>
<td>0.3 lb</td>
<td>1.5 lb</td>
<td>1.6 lb</td>
<td>6 lb</td>
<td>6.2 lb</td>
<td>8 lb</td>
</tr>
</tbody>
</table>

### SAFETY

<table>
<thead>
<tr>
<th></th>
<th>TS100</th>
<th>TS400</th>
<th>TS800</th>
<th>TS1000</th>
<th>TS2000</th>
<th>TS3000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approvals</td>
<td>FCC Class A, CE</td>
<td>FCC Class A, CE</td>
<td>FCC Class A, CE</td>
<td>FCC Class A, CE</td>
<td>FCC Class A, CE</td>
<td>FCC Class A, CE</td>
</tr>
</tbody>
</table>

This section has all the information you need to quickly and successfully purchase or build cables to the Cyclades-TS. It focuses on information related to the RS-232 interface, which applies not only to the Cyclades-TS but also to any RS-232 cabling. At the end of this chapter you will also find some information about the RS-485 interface, which is available in the Cyclades-TS100 model only.

### The RS-232 Standard

RS-232C, EIA RS-232, or simply RS-232 refer to a standard defined by the Electronic Industries Association in 1969 for serial communication. More than 30 years later, we have found more applications for this standard than its creators could have imagined. Almost all electronic devices nowadays have serial communication ports.

RS-232 was defined to connect Data Terminal Equipment, (DTE, usually a computer or terminal) to Data Communication Equipment (DCE, usually a modem):

\[
\text{DTE} \rightarrow \text{RS-232} \rightarrow \text{DCE} \rightarrow \text{communication line} \rightarrow \text{DCE} \rightarrow \text{RS-232} \rightarrow \text{DTE}
\]
RS-232 is now mostly being used to connect DTE devices directly (without modems or communication lines in between). While that was not the original intention, it is possible with some wiring tricks. The relevant signals (or wires) in a RS-232 cable, from the standpoint of the computer (DTE), are:

**Receive Data (RxD) and Transmit Data (TxD)** – The actual data signals

**Signal Ground (Gnd)** - Electrical reference for both ends

**Data Terminal Ready (DTR)** - Indicates that the computer (DTE) is active

**Data Set Ready (DSR)** - Indicates that the modem (DCE) is active.

**Data Carrier Ready (DCD)** - Indicates that the connection over the communication line is active

**CTS (Clear to Send, an input)** – Flow control for data flowing from DTE to DCE

**RTS (Request to Send, an output)** – Flow control for data flowing from DCE to DTE

Not all signals are necessary for every application, so the RS-232 cable may not need all 7 wires. The RS-232 interface defines communication parameters such as parity, number of bits per character, number of stop-bits and the baud rate. Both sides must be configured with the same parameters. That is the first thing to verify if you think you have the correct cable and things still do not work. The most common configuration is 8N1 (8 bits of data per character, no parity bit included with the data, 1 stop-bit to indicate the end of a character).

The baud rate in a RS-232 line translates directly into the data speed in bits per second (bps). Usual transmission speeds range between 9,600 bps and 19,200bps (used in most automation and console applications) to 115,200 bps (used by the fastest modems).

**Cable Length**

The original RS-232 specifications were defined to work at a maximum speed of 19,200 bps over distances up to 15 meters (or about 50 feet). That was 30 years ago. Today, RS-232 interfaces can drive signals faster and through longer cables.

As a general rule, consider:

- If the speed is lower than 38.4 kbps, you are safe with any cable up to 30 meters (100 feet)
- If the speed is 38.4 kbps or higher, cables should be shorter than 10 meters (30 feet)
If your application is outside the above limits (high speed, long distances), you will need better quality (low-impedance, low-capacitance) cables. Successful RS-232 data transmission depends on many variables that are specific to each environment. The general rules above are empirical and have a lot of safety margins built-in.

**Connectors**

The connector traditionally used with RS-232 is the 25-pin D-shaped connector (DB-25). Most analog modems and most older computers and serial equipment use this connector. The RS-232 interface on DB-25 connector always uses the same standard pin assignment.

The 9-pin D-shaped connector (DB-9) saves some space and is also used for RS-232. Most new PC COM ports and serial equipment (specially when compact size is important) uses this connector. RS-232 interfaces on DB-9 connectors always use the same standard pin assignment.

The telephone-type modular RJ-45 plug and jack are very compact, inexpensive and compatible with the phone and Ethernet wiring systems present in most buildings and data centers. Most networking equipment and new servers use RJ-45 connectors for serial communication. Unfortunately there is no standard RS-232 pin assignment for RJ-45 connectors. Every equipment vendor has its pin assignment.

Most connectors have two versions. The ones with pins are said to be “male” and the ones with holes are said to be “female”.

<table>
<thead>
<tr>
<th>RS-232 Signal</th>
<th>Name/Function (Input/Output)</th>
<th>DB-25 pins (Standard)</th>
<th>DB-9 pins (Standard)</th>
<th>RJ-45 pins (Cyclades)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis</td>
<td>Safety Ground</td>
<td>1</td>
<td>Shell</td>
<td>Shell</td>
</tr>
<tr>
<td>TxD</td>
<td>Transmit Data (O)</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>RxD</td>
<td>Receive Data (I)</td>
<td>3</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>DTR</td>
<td>Data Terminal Ready (O)</td>
<td>20</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>DSR</td>
<td>Data Set Ready (I)</td>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>DCD</td>
<td>Data Carrier Detect (I)</td>
<td>8</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>RTS</td>
<td>Request To Send (O)</td>
<td>4</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send (I)</td>
<td>5</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Gnd</td>
<td>Signal Ground</td>
<td>7</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

**Straight-Through vs. Crossover Cables**

The RS-232 interface was originally intended to connect a DTE (computer, printer and other serial devices) to a DCE (modem) using a straight-through cable (all signals on one side connecting to the corresponding signals on the other side one-to-one). By using some “cabling tricks”, we can use RS-232 to connect two DTEs as is the case in most modern applications.

A crossover (a.k.a. null-modem) cable is used to connect two DTEs directly, without modems or communication lines in between. The data signals between the two sides are transmitted and received and there are many variations on how the other control signals are wired. A “complete” crossover cable would connect TxD with RxD, DTR with DCD/DSR, and RTS with CTS on both sides. A “simplified” crossover cable would cross TxD and RxD and locally short-circuit DTR with DCD/DSR and RTS with CTS.

**Which Cable Should be Used**

First, look up the proper cable for your application in the table below. Next, purchase standard off-the-shelf cables from a computer store or cable vendor. For custom cables, refer to the cable diagrams to build your own cables or order them from Cyclades or a cable vendor.
<table>
<thead>
<tr>
<th>To Connect To</th>
<th>Use Cable</th>
</tr>
</thead>
</table>
| DCE DB-25 Female (standard)                       | Cable 1 – RJ-45 to DB-25 M straight-through (Custom)  
This custom cable can be ordered from Cyclades or other cable vendors. A sample is included with the product ("straight-through"). |
| - Analog Modems                                   |                                                                                                                                         |
| - ISDN Terminal Adapters                          |                                                                                                                                         |
| DTE DB-25 Male or Female (standard)               | Cable 2 – RJ-45 to DB-25 F/M crossover (Custom)  
This custom cable can be ordered from Cyclades or other cable vendors. A sample is included with the products ("Console"). |
| - Serial Terminals                                |                                                                                                                                         |
| - Old PC COM ports                                |                                                                                                                                         |
| - Most serial printers                            |                                                                                                                                         |
| - Some Console Ports                              |                                                                                                                                         |
| - Most automation devices                         |                                                                                                                                         |
| DTE DB-9 Male or Female (standard)                | Cable 3 – RJ-45 to DB-9 F/M crossover (custom)  
This custom cable can be ordered from Cyclades or other cable vendors. A sample is included with the products ("Console"). |
| - Newer PC COM ports                              |                                                                                                                                         |
| - Most Mice and pointing devices                  |                                                                                                                                         |
| - Some automation devices                         |                                                                                                                                         |
| DTE RJ-45 Cyclades (custom)                       | Cable 4 – RJ-45 to RJ-45 crossover (custom)  
This custom cable can be ordered from Cyclades or cable vendors using the provided wiring diagram. |
| - All Cyclades Console Ports                      |                                                                                                                                         |
| DTE RJ-45 Netra (custom)                          | Cable 5 – RJ-45 to RJ-45 crossover (custom)  
This custom cable can be ordered from Cyclades or cable vendors using the provided wiring diagram. |
| - Sun Netra Console Ports                         |                                                                                                                                         |
| - Cisco Console Ports                             |                                                                                                                                         |

**Cable Diagrams**

Before using the following cable diagrams refer to the tables above to select the correct cable for your application. Sometimes, crossover cables are wired slightly differently depending on the application. A “complete” crossover
cable would connect the TxD with RxD, DTR with DCD/DSR, and RTS with CTS across both sides. A “simplified” crossover cable would cross TxD and RxD and locally short-circuit DTR with DCD/DSR and RTS with CTS.

Most of the diagrams in this document show the “complete” version of the crossover cables, with support for modem control signals and hardware flow control. Applications that do not require such features have just to configure NO hardware flow control and NO DCD detection on their side. Both ends should have the same configuration for better use of the complete version of the cables.

**Cable #1: Cyclades RJ-45 to DB-25 Male, Straight Through**

Application: It connects Cyclades products (serial ports) to modems and other DCE RS-232 devices.

```
<table>
<thead>
<tr>
<th>RJ-45 Male</th>
<th>DB-25 Male</th>
</tr>
</thead>
<tbody>
<tr>
<td>TxD 3</td>
<td>TxD 2</td>
</tr>
<tr>
<td>RxD 6</td>
<td>RxD 3</td>
</tr>
<tr>
<td>Gnd 4</td>
<td>Gnd 7</td>
</tr>
<tr>
<td>DTR 2</td>
<td>DTR 20</td>
</tr>
<tr>
<td>DSR 8</td>
<td>DSR 6</td>
</tr>
<tr>
<td>DCD 7</td>
<td>DCD 8</td>
</tr>
<tr>
<td>RTS 1</td>
<td>RTS 4</td>
</tr>
<tr>
<td>CTS 5</td>
<td>CTS 5</td>
</tr>
</tbody>
</table>
```
**Cable #2: Cyclades RJ-45 to DB-25 Female/Male, Crossover**

Application: It connects Cyclades products (serial ports) to console ports, terminals, printers and other DTE RS-232 devices.

<table>
<thead>
<tr>
<th>RJ-45 Custom</th>
<th>DB-25 F/M</th>
</tr>
</thead>
<tbody>
<tr>
<td>TxD 3</td>
<td>RxD 3</td>
</tr>
<tr>
<td>RxD 6</td>
<td>TxD 2</td>
</tr>
<tr>
<td>Gnd 4</td>
<td>Gnd 7</td>
</tr>
<tr>
<td>DTR 2</td>
<td>DSR 6</td>
</tr>
<tr>
<td>DSR 8</td>
<td>DCD 6</td>
</tr>
<tr>
<td>DCD 7</td>
<td>DTR 8</td>
</tr>
<tr>
<td>RTS 1</td>
<td>RTS 4</td>
</tr>
<tr>
<td>CTS 5</td>
<td>CTS 5</td>
</tr>
</tbody>
</table>

![Diagram of RJ-45 to DB-25 Female/Male Crossover Cable](image)
**Cable #3: Cyclades RJ-45 to DB-9 Female, Crossover**

Application: It connects Cyclades products (serial ports) to console ports, terminals, printers and other DTE RS-232 devices.

```
RJ-45
Cross
```

<table>
<thead>
<tr>
<th>RJ-45</th>
<th>DB-9 Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom</td>
<td></td>
</tr>
<tr>
<td>TxD 3</td>
<td>RxD 2</td>
</tr>
<tr>
<td>RxD 6</td>
<td>TxD 3</td>
</tr>
<tr>
<td>Gnd 4</td>
<td>Gnd 5</td>
</tr>
<tr>
<td>DTR 2</td>
<td>DSR 6</td>
</tr>
<tr>
<td>DSR 8</td>
<td>DCD 1</td>
</tr>
<tr>
<td>DCD 7</td>
<td>DTR 4</td>
</tr>
<tr>
<td>RTS 1</td>
<td>CTS 8</td>
</tr>
<tr>
<td>CTS 5</td>
<td>RTS 7</td>
</tr>
</tbody>
</table>

```
DB-9 Female
```

```
RJ-45
```
Cable #4: DB-9 Female to DB-25 Female, Crossover

Application: It connects the Cyclades-TS100 (serial port) to terminals, printers and other DTE RS-232 devices.

<table>
<thead>
<tr>
<th>DB-9 Female</th>
<th>DB-25 Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxD 2</td>
<td>2 TxD</td>
</tr>
<tr>
<td>TxD 3</td>
<td>3 RxD</td>
</tr>
<tr>
<td>Gnd 5</td>
<td>7 Gnd</td>
</tr>
<tr>
<td>DSR 6</td>
<td>20 DTR</td>
</tr>
<tr>
<td>DCD 1</td>
<td>6 DsR</td>
</tr>
<tr>
<td>DTR 4</td>
<td>8 DCD</td>
</tr>
<tr>
<td>RTS 7</td>
<td>5 CTS</td>
</tr>
<tr>
<td>CTS 8</td>
<td>4 RTS</td>
</tr>
</tbody>
</table>
**Cable #5: Cyclades RJ-45 to Cyclades RJ-45, Crossover**

Application: Usually used to connect two ports of a Cyclades product ("loopback") for testing purposes.

- **RJ-45 Male**
  - TxD 3
  - RxD 6
  - Gnd 4
  - DTR 2
  - DSR 8
  - DCD 7
  - RTS 1
  - CTS 5

- **RJ-45 Male**
  - TxD 3
  - RxD 6
  - Gnd 4
  - DTR 2
  - DSR 8
  - DCD 7
  - RTS 1
  - CTS 5
Cable #6: Cyclades RJ-45 to Netra RJ-45, Crossover

Usually used in console management applications to connect Cyclades products to a Sun Netra server or to a Cisco product.

<table>
<thead>
<tr>
<th>RJ-45 Custom</th>
<th>RJ-45 Netra</th>
</tr>
</thead>
<tbody>
<tr>
<td>TxD 3</td>
<td>RxD 6</td>
</tr>
<tr>
<td>RxD 6</td>
<td>TxD 3</td>
</tr>
<tr>
<td>Gnd 4</td>
<td>Gnd 4</td>
</tr>
<tr>
<td>DTR 2</td>
<td>DSR 7</td>
</tr>
<tr>
<td>DCD 7</td>
<td>DTR 2</td>
</tr>
<tr>
<td>RTS 1</td>
<td>CTS 8</td>
</tr>
<tr>
<td>CTS 5</td>
<td>RTS 1</td>
</tr>
</tbody>
</table>
Loop-Back Connector for Hardware Test
The use of the following DB-25 connector is explained in the Troubleshooting chapter.

DB-25 Male to DB-9 Female Adapter
The following adapter may be necessary.
Cabling Information Applicable only to the TS100

The RS-485 Standard
The RS-485 is another standard for serial communication and is available only in the Cyclades-TS100. Different from the RS-232, the RS-485 uses fewer wires - either two wires (one twisted pair) for half duplex communication or four wires (two twisted pairs) for full duplex communication. Another RS-485 characteristic is the “termination”. In a network that uses the RS-485 standard, the equipments are connected one to the other in a cascade arrangement. A “termination” is required from the last equipment to set the end of this network.

TS100 Connectors
Although the RS-485 can be provided in different kinds of connectors, the Cyclades-TS100 uses a 9-pin D-shaped connector (DB-9) and a block connector with the pin assignment described below.

<table>
<thead>
<tr>
<th>RS-485 Signal</th>
<th>Name/Function</th>
<th>DB-9 pins</th>
<th>Block connector pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis</td>
<td>Safety Ground</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>TXD-</td>
<td>Transmit Data - (A)</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>TXD+</td>
<td>Transmit Data + (B)</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>RXD+</td>
<td>Receive Data + (B)</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>RXD-</td>
<td>Receive Data - (A)</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Chassis</td>
<td>Safety Ground</td>
<td></td>
<td>6</td>
</tr>
</tbody>
</table>

Notice that if the Cyclades-TS100 is configured to use RS-485, the RS-485 signals will be available in both DB-
9 and block connector. In this case, the DB-9 pins used in an RS-232 connection can be considered not connected.

**Cable diagrams**

**Cable #1: DB-9 Female to DB-9 Female, Crossover half duplex**
Application: It connects the Cyclades-TS100 (serial port) DTE RS-485 devices with half duplex communication.

<table>
<thead>
<tr>
<th>DB-9 Female</th>
<th>DB-9 Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxD -8</td>
<td></td>
</tr>
<tr>
<td>TxD -7</td>
<td>RxD +2</td>
</tr>
<tr>
<td></td>
<td>TxD +3</td>
</tr>
</tbody>
</table>

**Cable #2: DB-9 Female to DB-9 Female, Crossover full duplex**
Application: It connects the Cyclades-TS100 (serial port) to DTE RS-485 devices with full duplex communication.

<table>
<thead>
<tr>
<th>DB-9 Female</th>
<th>DB-9 Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxD -8</td>
<td>TxD -7</td>
</tr>
<tr>
<td>TxD -7</td>
<td>RxD +2</td>
</tr>
<tr>
<td>RxD +2</td>
<td>TxD +3</td>
</tr>
<tr>
<td></td>
<td>RxD +2</td>
</tr>
</tbody>
</table>
**Cable #3: Block Connector to Block Connector, Crossover half duplex**
Application: It connects the Cyclades-TS100 (serial port) to DTE RS-485 devices with half duplex communication.

```
<table>
<thead>
<tr>
<th></th>
<th>Block Connector</th>
<th>Block Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxD -5</td>
<td></td>
<td>RxD -5</td>
</tr>
<tr>
<td>TxD -2</td>
<td></td>
<td>TxD -2</td>
</tr>
<tr>
<td>RxD +4</td>
<td></td>
<td>RxD +4</td>
</tr>
<tr>
<td>TxD +3</td>
<td></td>
<td>TxD +3</td>
</tr>
</tbody>
</table>
```

**Cable #4: Block Connector to Block Connector, Crossover full duplex**
Application: It connects the Cyclades-TS100 (serial port) to DTE RS-485 devices with full duplex communication.

```
<table>
<thead>
<tr>
<th></th>
<th>Block Connector</th>
<th>Block Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxD -5</td>
<td></td>
<td>TxD -2</td>
</tr>
<tr>
<td>RxD +4</td>
<td></td>
<td>TxD +3</td>
</tr>
<tr>
<td>TxD -2</td>
<td></td>
<td>RxD -5</td>
</tr>
<tr>
<td>TxD +3</td>
<td></td>
<td>RxD +4</td>
</tr>
</tbody>
</table>
```
APPENDIX C  SAMPLE PSLAVE.CONF FILES

The pslave.conf file with all possible parameters and their descriptions is presented first. The pslave.conf files for the three examples configured in chapter 6 follow.

The Complete pslave.conf File Provided with the Cyclades-TS

```
# pslave.conf Sample server configuration file.
#
# The Terminal Server uses a virtual terminal concept. Virtual terminals are
# named s1, s2, etc. Every virtual terminal should have a related
# physical device tty (without the "/dev/"). The tty parameter
# must be configured and must be unique for each virtual terminal.
#
# There two types of parameters:
# 1) Global parameters
#    These parameters have the prefix "conf." Example of global parameters
#    are ethernet ip address, etc.
# 2) Terminal Parameters.
#    These parameters have prefixes "all.", "s1.", "s2.", etc.
#
# The "all." entries are used as a template for all virtual terminals.
# Setting all.speed to 9600 will set all virtual terminal (s1, s2, s3, etc.) speeds to 9600.
# Note that you can change the "all." settings one by one.
# If the parameter "s4.speed 19200" appears later in the file, all
terminals except s4 will have speed 9600 bps and "s4" will have speed 19200 bps.
#
# Expansion Variables
# A list of format strings used by some parameters is provided here
# for reference.
#
# %l: login name
```
# %L: stripped login name
# %p: NAS port number
# %P: protocol
# %b: port speed
# %i: local IP
# %j: remote IP
# %1: first byte (MSB) of remote IP
# %2: second byte of remote IP
# %3: third byte of remote IP
# %4: fourth (LSB) byte of remote IP
# %c: connect-info
# %m: netmask
# %t: MTU
# %r: MRU
# %I: idle timeout
# %T: session timeout
# %h: hostname
# %\: \%

# Generic SAMPLE:
# all async ports at 9600 bps, 8N1, no flow control
# Eth IP address 192.169.160.10/24 (MTU=1500)
# protocol socket_server
# host IP 192.168.160.8/24
# Radius Server IP 192.168.160.3 (authentication and accounting)
# authentication none

# Ethernet configuration.
# These parameters should only be configured in the file
# /etc/network/ifcfg_eth0 _IF_ the customer will not be using the
cy_ras/portslave applications. If the cy_ras/portslave applications are _NOT_
# used put all ifconfig commands for the ethernet directly in the
# /etc/network/ifcfg_eth0.
# The cy_ras application OVERWRITES the ifcfg_eth0 file with the
# values configured here.
# The Cyclades-TS can request all of its ethernet parameters to a DHCP server.
# The administrator can activate the dhcp client with more options changing
# the file /etc/network/dhcpcd_cmd.
#
# Valid values 0: DHCP disabled (default)
# 1: DHCP active
# 2: DHCP active and the TS saves in flash the last ip assigned
# by the DHCP server. This option requires changes in the
# files /etc/config_files and /etc/network/dhcpcd_cmd
# #
# SEE Cyclades-TS manual for more information.
# #
#conf.dhcp_client 1
conf.eth_ip 192.168.160.10
conf.eth_mask 255.255.255.0
conf.eth_mtu 1500

# # Secondary IP address of ethernet #
#
#conf.eth_ip_alias 192.168.161.10
#conf.eth_mask_alias 255.255.255.0

# # Remote Network File System where data buffering will be written instead
# of the default directory '/var/run/DB'. The directory tree to which the
# file will be written must be NFS-mounted.
# # If data buffering is turned on for port 1, for example, the data will be
# stored in /tmp/ts_data_buffer/{ttyS1.data | serverfarm} on the machines
# with IP address 192.168.160.11. The remote host must have NFS installed
# and the administrator must create, export and allow reading/writing to
# this directory.
# The size of this file is not limited by the value of the parameter
# sl.data_buffering, though the value cannot be zero since a zero value turns
# off data buffering.
# #conf.nfs_data_buffering 192.168.160.11:/tmp/ts_data_buffer

# # Lock directory - The lock directory is /var/lock for the Cyclades-TS.
# It should not be changed unless the user decides to customize the
# operating system.
# conf.lockdir/var/lock
# Location of the rlogin binary that accepts the "-i" flag.
# conf.rlogin /usr/local/bin/rlogin-radius
# Location of our patched pppd with Radius linked in.
# conf.pppd /usr/local/sbin/pppd
# Location of the telnet utility. This can be the system telnet. (Optional)
# conf.telnet /bin/telnet
# Location of ssh utility. This can be the system SSH. (Optional)
# conf.ssh /bin/ssh
# This parameter is only necessary when authentication is being
# performed for a port. When set to one, it is possible to log
# in to the Terminal Server directly
# by placing a "!" before your login name, then using your normal
# password. This is useful if the Radius authentication server is down.
# conf.locallogins 1
#
#
# Syslog facility for portslave
# conf.facility 7
#
# Syslog facility for Data Buffering and Alarm
# conf.DB_facility 7
#
# User groups make the configuration of Port access restrictions
# easier. The parameter s<nn>.users, that will be explained later,
# can be configured using a combination of group names and user names.
#conf.group mkt: paul, sam
#conf.group adm: joe, mark
#s1.users mkt, joe
#s2.users adm, sam

# Speed. All ports are set to 9600 baud rate, 8 bits, No parity, 1 stop bit.
# These values can be changed port by port later in the file.
all.speed 9600
all.datasize 8
all.stopbits 1
all.parity none

# Media type - define media type and operation mode (half/full) duplex.
# valid values:
# rs232 - RS232 (default value).
# rs485_half - RS485 half duplex without terminator
# rs485_full - RS485 full duplex without terminator
# rs485_half_terminator - RS485 half duplex with terminator
# rs485_full_terminator - RS485 full duplex with terminator
# rs422 - alike rs485_full
# rs422_terminator - alike rs485_full_terminator

#all.media rs232

# Authentication type - either "local", "radius", "none", "remote" # "local/radius", "radius/local", or "RadiusDownLocal".
# If the authentication type is configured as "local/radius" the pslave
# first tries to authenticate locally. If it fails, pslave will try to
# authenticate using the radius server.
# If the authentication type is configured as "RadiusDownLocal" the portslave
# first tries to authenticate using the radius server. If the Radius server
# sends back a rejection, authentication will fail. Local authentication
# will be tried only if the Radius server is down (timeout).
#
all.authtype none
#
# Authentication host and accounting host. Two of each can be configured
# per port. The first is tried 'radretries' times before the
# second is tried. If 'radretries' is not configured, 5 is used by default.
# The parameter 'radtimeout' sets the timeout per query in seconds.
#
all.authhost1 192.168.160.3
all.accthost1 192.168.160.3
all.radtimeout 3
all.radretries 5

# The shared secret used by RADIUS.
#
all.secret cyclades
#
# Default protocol.
#
# Valid values are
# RAS profile: "slip", "cslip", "ppp", "ppp_only"
# TS profile: "login", "rlogin", "telnet", # "ssh", "ssh2", "socket_client"
# CAS profile: "socket_server", "socket_ssh", "raw_data"
# ppp_only ==> PPP over leased lines (only authentication PAP/CHAP)
# ppp ==> PPP with terminal post dialing (Auto detect PPP)
#
# Default ip address of linux host to which the terminals will connect.
# Used by the protocols rlogin, ssh, socket_client, etc.
#
all.host 192.168.160.8
# IP Address assigned to the serial port.
# The '+' after the value causes the interfaces to have
# consecutive ip addresses. Ex. 192.168.1.101, 192.168.1.107, etc.
# The IP number of a port is used when the RADIUS
# server does not send an IP number, or if it tells us to use a dynamic IP no.
all.ipno 192.168.1.101+
all.netmask 255.255.255.255

# Maximum reception/transmission unit size for the port
all.mtu 1500
all.mru 1500

# Standard message issued on connect.
all.issue \r\n\nTSLINUX - Portslave Internet Services\n\r\nWelcome to terminal server %h port S%p \n\r\nCustomer Support: 510-770-9727 http://www.cyclades.com/\n\r\n
# Login prompt.
all.prompt %h login:

# Terminal type, for rlogin/telnet sessions.
all.term vt100

# If you want the Terminal Server to update the
# login records (written to the /var/run/utmp and/or /var/log/wtmp
# files), set sysutmp/syswtmp to 1. This is useful for tracking
# who has accessed the Terminal Server and what they did.
# all.sysutmp 1
# all.syswtmp 0
# all.utmpfrom  "%p:%P.%3.%4"

# Use initchat to initialize the modem.
# d == delay (1 sec), p == pause (0.1 sec), l == toggle DTR
# r == <CR>, l == <LF>
# all.initchat  TIMEOUT 10 \ 
#     "\d\l\dATZ \ 
#     OK\r\n-ATZ-OK\r\n " \ 
#     TIMEOUT 10 \ 
#     "" ATA \ 
#     OK\r\n " \ 
#     TIMEOUT 3600 \ 
#     RING " " \ 
#     STATUS Incoming %p:I.HANDSHAKE \ 
#     "" ATA \ 
#     TIMEOUT 60 \ 
#     CONNECT@ " " \ 
#     STATUS Connected %p:I.HANDSHAKE

# Serial port flow control:
#     hard - hardware, rts/cts
#     soft - software, CTRL-S / CTRL-Q
#     none.
# all.flow    none

# DCD signal (sets the tty parameter CLOCAL). Valid values are 0 or 1.
#     In a socket session, if all.dcd=0, a connection request (telnet or
#     ssh) will be accepted regardless of the DCD signal and the connection
#     will not be closed if the DCD signal is set to DOWN.
#     In a socket connection, if all.dcd=1 a connection request will be
#     accepted only if the DCD signal is UP and the connection (telnet or
#     ssh) will be closed if the DCD signal is set to DOWN.
# all.dcd 0

# PPP options - used if a PPP session is autodetected.
# Note that mru and mtu are both set to the MTU setting.
# Callback server is enabled when cb-script parameter is set.
# all.autopp%i:%j novj
# proxyarp modem asyncmap 00A0000
# noipx noccpp login auth require-pap refuse-chap
# mtu %t mru %t
# ms-dns 192.168.160.5 ms-dns 0.0.0.0
# cb-script /etc/pslave/cb_script
# plugin /usr/lib/libpsr.so

# PPP options - User already authenticated and service type is PPP.
# all.pppopt %i:%j novj
# proxyarp modem asyncmap 00A0000
# noipx noccpp mtu %t mru %t netmask %m
# idle %I maxconnect %T
# ms-dns 192.168.160.5 ms-dns 0.0.0.0
# plugin /usr/lib/libpsr.so

# When not set to zero, this parameter sets the wait for a TCP connection
# keep-alive timer. If no traffic passes through the Terminal Server for
# this period of time (ms), the Terminal Server will send a modem status
# message to the remote device to see if the connection is still up.
# all.poll_interval 1000

# Transmission interval - Controls the interval between two consecutive datas
# packets transmitted to the Ethernet. Only valid for
# protocols socket_server, raw_data, and
# socket_client.
# Valid values : 0 - transmit packet immediately (no interval).
# 10, 20, 30, ... interval in milliseconds.
# all.tx_interval 100
#
# Inactivity timeout - Defines the time in minutes that a connection can
# remain without activity (rx/tx). Only for CAS profile
# and socket_client protocol.
# all.idletimeout 5
#
# This defines an alternative labeling system for the Terminal Server ports.
# This parameter is used by the protocols telnet, socket_client and
# socket_server. It is mandatory if the protocol is socket_server, otherwise
# 23 will be used.
# The '+' after the numerical value causes the interfaces to be numbered
# consecutively. Ex. 7001, 7002, 7003, etc.
# all.socket_port 7001+
#
# Data buffering configuration
#
# A non-zero value activates data buffering. The number is equal to the
# buffer size. A file /var/run/DB/(ttyS#.data | ... available ram disk, NFS_buffering can be used.  There
# is effectively no limit to NFS buffer size.
# all.data_buffering 0
#
# When non-zero, the contents of the data buffer are sent to the syslog
# server every time a quantity of data equal to this parameter is collected.
# [40 to 255 recommended]
all.syslog_buffering 0

# Alarm configuration
# When non zero, all data received from the port is captured and is sent to syslog-ng
# with LOCAL [0+DB_facility] facility and INFO level.
# The syslog-ng.conf file should be set accordingly to make an action
# (please see the documentation).
all.alarm 0

# Controls the presentation of the Data buffering menu
# MENU:
# "A non-empty Data Buffering File was found. Choose which action
# should be performed ( (I)gnore, (D)isplay, (E)rase or (S)how and erase ) :"
# valid values:
# 0 - Shows the menu with all options.
# 1 - Doesn't show the menu and any non empty data buffering file
# 2 - Doesn't show the menu but shows a non empty data buffering file
# 3 - Shows the menu without the options "erase" and "show and erase".
all.dont_show_DBmenu 1

# Send Break to the TTY when this string is received (ssh only).
all.break_sequence ~break

# Authentication of Radius users registered without passwords
# When enabled (value 1) and a user registered in
# the Radius database with a blank password tries to log in, the user
# is authenticated. This is a very weak level of security since
# a user would only need to know that a particular username exists.
# This does not affect Radius users registered with passwords.
all.radnullpass 0
# Automatic User Definition (more useful when used to a specific port)
# This parameter is only used if the port is configured as a Terminal Server
# (login, telnet, rlogin, ssh and ssh2) and authentication type 'none'.
# all.userauto edson

# Port access restriction (more useful when used to a specific port).
# A single comma and spaces/tabs may be used between names.
# A comma may not appear between the ! and the first user name.
# The users may be local or Radius.
# In this example, the users joe and mark CANNOT access any serial port
# all.users ! joe, mark
# In this example, ONLY the users joe and mark CAN access any serial port
# all.users  joe, mark

# Serverfarm is an alias name for a server connected to the Cyclades-TS
# through one of its serial ports (only useful if assigned to a specific port).
# This alias is used as name to the data buffering file and in ssh command to
# select a serial port that should be configured as "socket_ssh".
# The value entered here should be the same used in the ssh command. Ex.
# ssh -t <username>:@<server_connected_to_serial1>@<tsname> or
# ssh -t -l <username>:<server_connected_to_serial1> <tsname>
# s1.serverfarm server_connected_to_serial1

# Snif session mode (in, out, i/o). With this parameter the user can select
# which data will be sent to the monitor. The default is "out".
# all.sniff_mode out

Appendix C - Sample pslave.conf files-The complete pslave.conf file
# Users that are allowed to sniff sessions (administrator). This field has
# the same format "all.users", but the '!' should be used used with
# PRECAUTION.
#
# In this example, ONLY the users joe, mark, and peter CAN access any
# serial port (to create first session) but ONLY the user peter can
# sniff or cancel another session.
# all.users   joe, mark
# all.admin_users peter

#
# Port-specific parameters
#
# s1.tty       ttyS1
# s2.tty       ttyS2
# s3.tty       ttyS3
# s4.tty       ttyS4
# s5.tty       ttyS5
# s6.tty       ttyS6
# s7.tty       ttyS7
# s8.tty       ttyS8
# s9.tty       ttyS9
# s10.tty      ttyS10
# s11.tty      ttyS11
# s12.tty      ttyS12
# s13.tty      ttyS13
# s14.tty      ttyS14
# s15.tty      ttyS15
# s16.tty      ttyS16

# for TS2000 uncomment s17 through s32
#s17.tty       ttyS17
#s18.tty       ttyS18
#s19.tty       ttyS19
#s20.tty       ttyS20
#s21.tty       ttyS21
#s22.tty       ttyS22
#s23.tty       ttyS23
#s24.tty       ttyS24
#s25.tty       ttyS25
#s26.tty  ttyS26
#s27.tty  ttyS27
#s28.tty  ttyS28
#s29.tty  ttyS29
#s30.tty  ttyS30
#s31.tty  ttyS31
#s32.tty  ttyS32

# for TS3000 uncomment s33 through s48
#s33.tty  ttyS33
#s34.tty  ttyS34
#s35.tty  ttyS35
#s36.tty  ttyS36
#s37.tty  ttyS37
#s38.tty  ttyS38
#s39.tty  ttyS39
#s40.tty  ttyS40
#s41.tty  ttyS41
#s42.tty  ttyS42
#s43.tty  ttyS43
#s44.tty  ttyS44
#s45.tty  ttyS45
#s46.tty  ttyS46
#s47.tty  ttyS47
#s48.tty  ttyS48
The pslave.cas File Provided With the Cyclades-TS for the Console Access Server Example

```
# pslave.conf Sample server configuration file.
#
# Console Access Server Profile
#

conf.eth_ip     200.200.200.1
conf.eth_mask   255.255.255.0
conf.eth_mtu1500
conf.nfs_data_buffering 192.168.160.11:/tmp/ts_data_buffer
conf.lockdir/var/lock
conf.facility 7

all.speed       9600
all.datasize    8
all.stopbits    1
all.parity      none
all.authtype    radius
all.authhost1   200.200.200.2
all.accthost1   200.200.200.2
all.radtimeout  3
all.radretries  5
all.secret      cyclades
all.ipno        192.168.1.101+
all.term        vt100
all.issue       \r\n
TS LINUX - Portslave Internet Services

Welcome to terminal server %h port $p 


all.prompt  %h login:
all.term    vt100
all.flow    hard
all.poll_interval  0
all.socket_port 7001+
all.protocol socket_server
```
all.data_buffering 0
all.syslog_buffering 0
#all.dont_show_DBmenu 1

# Users joe and mark will only have access granted to the serial port ttyS2
all.users ! joe, mark

# Sniff sessions will only display data sent by servers connected
to the serial port.
all.sniff_mode out

# Only users peter and john can open a sniff session
all.admin_users peter, john

# Port-specific parameters
#
#-----------------
# PORT 1
#-----------------

s1.tty   ttyS1
s1.authtype   local
s1.serverfarm server_connected_serial1

#-----------------
# PORT 2
#-----------------

s2.tty   ttyS2
s2.users   joe, mark
s2.protocol   socket_ssh

#-----------------
# PORT 8
#-----------------
s8.tty     ttyS8
s8.protocol socket_ssh
s8.authtype none
s8.serverfarm server_connected_serial8
The `pslave.ts` File provided with the Cyclades-TS for the Terminal Server Example

```conf
# pslave.conf Sample server configuration file.
#
# Terminal Server Profile

conf.eth_ip     200.200.200.1
conf.eth_mask   255.255.255.0
conf.eth_mtu1500
conf.lockdir/var/lock
conf.rlogin /usr/local/bin/rlogin-radius
conf.telnet /bin/telnet
conf.ssh       /bin/ssh
conf.locallogins 0

all.speed       9600
all.datasize    8
all.stopbits    1
all.parity      none
all.authtype    none
all.protocol    telnet
all.host        200.200.200.3
all.issue       \r\n
\r\n
TS LINUX - Portslave Internet Services\n
\r\n
Welcome to terminal server %h port S%p \n
\r\n
Customer Support: 510-770-9727 http://www.cyclades.com/\n\r\n
all.prompt     %h login:
all.term       vt100
all.flow       hard
all.socket_port 23

#
# Users joe and mark will only have access to serial port ttyS5
#
all.users ! joe, mark
```
# Port-specific parameters

s1.tty   ttyS1
s2.tty   ttyS2
s2.authtype  local
s2.protocol  rlogin
s2.speed     19200
s2.datasize  7
s2.stopbits  2
s2.parity    even

s3.tty   ttyS3
s3.protocol  ssh2
s3.authtype  remote

s4.tty   ttyS4
s4.protocol  ssh
s4.authtype  remote

s5.tty   ttyS5
s5.users  joe, mark
The **pslave.ras** File Provided With the Cyclades-TS for the Remote Access Server Example

```conf
# pslave.conf Sample server configuration file.
#
# Remote Access Server Profile
#
conf.eth_ip 200.200.200.1
conf.eth_mask 255.255.255.0
conf.eth_mtu1500
conf.lockdir/var/lock
conf.pppd /usr/local/sbin/pppd-radius
conf.facility 7
all.speed 57600
all.datasync 8
all.stopbits 1
all.parity none
all.authtype radius
all.authhost1 200.200.200.2
all.accthost1 200.200.200.2
all.radtimeout 5
all.radretries 5
all.secret cocomero
all.protocolppp
all.ipno 200.200.200.11+
all.netmask 255.255.255.255
all.mtu 1500
all.mru 1500
all.issue \r\n\r\
   TSLINUX - Portslave Internet Services\n\r\n   Welcome to terminal server %h port %p \n\r\nCustomer Support: 510-770-9727 http://www.cyclades.com/\n\r\n
all.initchat TIMEOUT 10 \n"" \d\l\dATZ \nOK\r\n\n"" \
```
"" ATMO \\
OK\R\N "" \\
TIMEOUT 3600 \\
RING "" \\
STATUS Incoming %p:I.HANDSHAKE \\
"" ATA \\
TIMEOUT 60 \\
CONNECT@ "" \\
STATUS Connected %p:I.HANDSHAKE \\

all.flow hard
all.dcd 1
all.autoppp %i:%j novj \\
proxyarp modem asyncmap 000A0000 \\
noipx noccp login auth require-pap refuse-chap \\
mtu %t mru %t \\
plugin /usr/lib/libpsr.so

all.pppopt %i:%j novj \\
proxyarp modem asyncmap 000A0000 \\
noipx noccp mtu %t mru %t netmask %t \\
idle %I maxconnect %T \\
plugin /usr/lib/libpsr.so

# # Port-specific parameters
#

#-------------------------------
# PORT 1 PPP dial in with terminal post dialing
#-------------------------------
s1.tty ttyS1

#-------------------------------
# PORT 2 PPP dial in with terminal post dialing
#-------------------------------
s2.tty ttyS2
s2.authtype local/radius

#-------------------------------
# PORT 3 PPP Leased line
#-------------------------------------------
s3.tty ttyS3
s3.protocol ppp_only
s3.pppopt %i:%j novj \
  proxyarp modem asyncmap 000A0000 \
  noipx noccp login auth require-pap refuse-chap \
  mtu %t mru %t \
  plugin /usr/lib/libpsr.so
s3.initchat ""
s3.issue    ""
APPENDIX D  CUSTOMIZATION

Everything related to the Cyclades-TS can be traced back to two files: /etc/rc.sysinit and /etc/inittab. All Cyclades-TS application programs are started during boot by the init process. The related lines in the /etc/inittab file are listed below:

```plaintext
# System initialization.
::sysinit:/etc/rc.sysinit

# Single user shell
#console::respawn:/bin/sh < /dev/console > /dev/console 2> /dev/console
ttyS0::respawn:/sbin/getty -p ttyS0 ansi
::respawn:/sbin/cy_wdt_led wdt led

# Cyclades RAS
::once:/sbin/cron
::once:/sbin/snmpd
::once:/sbin/cy_buffering
::once:/sbin/cy_ras
::once:/sbin/sshd -f /etc/ssh/sshd_config
::once:/sbin/ex_ntpclient
::once:/bin/webs
::once:/bin/syslog-ng
::once:/bin/cy_alarm
::wait:/sbin/fwset restore
```

To customize the Cyclades-TS, change these lines or add others.

For instance, to disable WEB services, comment the line referring to webs as follows.

```plaintext
#:once:/bin/webs
```

If the /etc/inittab file is changed, edit the /etc/config_files file and add a line containing only “/etc/inittab”. Save the file and exit the editor. Save the new configuration by executing saveconf. Then, the Cyclades-TS should be turned off and then turned on again. This is necessary because the init program provided by Busybox, a tool that emulates rm, cp, etc., but uses much less space, does not support the option ‘q’.
Cyclades provides a development kit which allows changes to be made to the Cyclades-TS’s software. However, Cyclades does not provide free technical support for systems modified in this way. Any changes are the responsibility of the user.
APPENDIX E  MULTIPLE SNIFFING

Versions 1.3.2 and earlier

Cyclades-TS allows a maximum of 2 connections to each serial port, as follows:

- 1 common session: user can execute read and write commands to the tty port. Session can be established by a regular user or by an administrator.

- 1 sniffer session: user can execute only read commands, in order to monitor what is going on in the other (main) session. Session can only be established by an administrator, defined by the parameter all.admin_users or sN.admin_users in the file pslave.conf (exception: authentication none - anyone can open a sniffer).

The first connection always opens a common session. After the second connection has been established and the user is authenticated, the Cyclades-TS shows the following menu to the administrator user:

```
* * * * ttySN is being used by (<user_name>) !!!
*
1 - Assume the main session
2 - Initiate a sniff session
3 - Quit
Enter your option :
```

If the second user is not an administrator, his connection is automatically refused.

This description is valid for all of the available protocols (socket_server, socket_ssh or raw_data).
Versions 1.3.3 and later

Users will be able to open more than one common and sniff sessions at the same port. For this purpose, the following configuration items will be included in the file pslave.conf:

- all.multiple_sessions: valid for all the serial ports; must be “yes” or “no”. The default value is “no”.

- sN.multiple_sessions: valid only for port N; must be “yes” or “no”. If it is not defined, it will assume the value of all.multiple_sessions.

- all.escape_char: valid for all the serial ports; this parameter will be used to present the menus below to the user. Only characters from ‘^a’ to ‘^z’ (i.e. CTRL-A to CTRL-Z) will be accepted. If this parameter is not set in pslave.conf, or in case it contains an invalid value, regular sessions will not be allowed to return to the menu regardless to what is typed by the user, whereas sniffer sessions will present the menu only if users type <CTRL-Z>. In addition, regular sessions will only be allowed to see the menu if the protocol used is “socket_server” or “socket_ssh”.

- sN.escape_char: valid only for port N; this parameter will be used to present the menus below to the user. Only characters from ‘^a’ to ‘^z’ (i.e. CTRL-A to CTRL-Z) will be accepted. If it is not defined, it will assume the value of all.escape_char.

When no multiple sessions are allowed for one port, the behavior of the Cyclades-TS when someone connects to it will be as described for version 1.3.2 and earlier. Otherwise, it will be as follows:

a. The first user to connect to the port will open a common session.

b. From the second connection on, only admin users will be allowed to connect to that port. The Cyclades-TS will send the following menu to these administrators (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

If the user selects 1 - Initiate a regular session, he will share that serial port with the users that were previously connected. He will read everything that is received by the serial port, and will also be able to write to it.

If the user selects 2 - Initiate a sniff session, he will start reading everything that is sent and/or received by the serial port, according to the parameter all.sniff_mode or sN.sniff_mode (that can be in, out or i/o).

When the user selects 3 - Send messages to another user, the Cyclades-TS will send the user’s messages to all the sessions, but not to the tty port. Everyone connected to that port will see all the “conversation” that’s going on, as if they were physically in front of the console in the same room. These messages will be formatted as [Message from user/PID] <<message text goes here>> by the TS.

To inform the Cyclades-TS that the message is to be sent to the serial port or not, the user will have to use the menu.

If the administrator chooses the option 4 - Kill session(s), the Cyclades-TS will show him a list of the pairs PID/user_name, and he will be able to select one session typing its PID, or “all” to kill all the sessions.

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

Only for the administrator users: typing all.escape_char or sN.escape_char from the normal or sniff session or
“send message mode” will make the TS show the previous menu. If this parameter is not set in pslave.conf, or it contains an invalid value, the regular sessions will not be allowed to return to the menu, and the sniffer sessions will be able to do it typing <CTRL-Z>. In addition, the regular session will only be allowed to see the menu if the protocol used is “socket_server” or “socket_ssh”.
APPENDIX F  CONFIGURATION WIZARD

Using Wizard through CLI

The user has a choice to configure the Cyclades-TS using the standard vi editor. For those not familiar with the editor, there’s a way to pre-configure the unit (just basic configuration such as IP address of the Cyclades-TS) using the CLI. After that, they can continue configuring the unit through the WEB.

Once using the WEB, there’s a wizard button to set basic parameters for a given profile (CAS, TS or RAS) to speed up the configuration process. This is used by customers who want basic features for a given profile. Customers who want to explore the features for that profile (data buffering, session sniffing, etc) would use the WEB thoroughly.

The configuration wizard application is a quicker and easier way to configure the Cyclades-TS. The use of this application is recommended if you are not familiar with the vi editor or if you just want to do a quick configuration of the TS.

The command ‘wiz’ gets you started with some basic configuration. After executing this, you can then use the web configuration manager to continue configurations for the TS. The files that will be eventually modified if you decide to save to flash at the end of this application are:

1. /etc/hostname
2. /etc/hosts
3. /etc/resolv.conf
4. /etc/network/st Routes
5. /etc/ports Slave/pslave.conf

Type ‘wiz’ in your TS terminal.
** ok, let's get started! I need a few basic information on the system so that it can know where it is located within the network and it can know about its neighbor or its local environment.

Set to defaults? (y/n) [N] :

** FIGURE F.1 **

The default answer or value to any question is in the brackets. For figure F.1, either just hit ENTER to execute whatever is in between the brackets or type ‘n’ to NOT reset the current configurations to the Cyclades defaults or type ‘y’ to reset to Cyclades default configurations.

The configuration begins in the next screens. There are instructions on how to use the wizard on each screen. There is also an explanation of each parameter before asking for it. To use the rest of the ‘wiz’ application, follow the instructions:

The default or the current value for the parameter is displayed inside the brackets. Just hit ENTER if you are satisfied with the value in the brackets. If not, enter the appropriate parameter and press ENTER.

If at any time, you want to exit the wizard or you want to skip the rest of the configuration, press ESC. This will immediately display a summary of the current configuration for your verification before exiting the application.
**INSTRUCTIONS:**

You can:

1) Enter the appropriate information for your system and press ENTER or
2) Press ENTER if you are satisfied with the value within the brackets [ ] and want to go on to the next parameter or
3) Press ESC if you want to exit.

**HOSTNAME** - An alias for your system. This way you can always refer to the system by this name rather than its IP address.

Hostname[TSx000]:

![FIGURE F.2]
INSTRUCTIONS:
You can:
   1) Enter the appropriate information for your system and press ENTER or
   2) Press ENTER if you are satisfied with the value within the brackets [ ] and want to go on to the next parameter or
   3) Press ESC if you want to exit.

IP - The IP address of your system (on its Ethernet interface.) This is the address of the system within your network. See your network administrator to obtain a valid IP address for the system.

IP of your system[192.168.160.10]

**FIGURE F.3**
INSTRUCTIONS:
You can:
1) Enter the appropriate information for your system and press ENTER or
2) Press ENTER if you are satisfied with the value within the brackets [ ] and want to go on to the next parameter or
3) Press ESC if you want to exit.

DOMAIN NAME - A name that locates or identifies your organization within the Internet.

Domain name[mycompany.com]:

FIGURE F.4
INSTRUCTIONS:
You can:
1) Enter the appropriate information for your system and press ENTER or
2) Press ENTER if you are satisfied with the value within the brackets [ ] and want to go on to the next parameter or
3) Press ESC if you want to exit.

DOMAIN NAME SERVER - The IP address of the server that resolves domain names. Your domain name is alphabetic so that it is easier to remember. Everytime you see the domain name, it is actually being translated into an IP address by the domain name server. See your network administrator to obtain this IP address for the domain name server.

Domain Name Server[127.0.0.1]:

FIGURE F.5
INSTRUCTIONS:
You can:
1) Enter the appropriate information for your system and press ENTER or
2) Press ENTER if you are satisfied with the value within the brackets [ ] and want to go on to the next parameter or
3) Press ESC if you want to exit.

GATEWAY - A node on a network that serves as an entrance point into another network. See your network administrator to find out your organization's gateway address.

Gateway IP[192.168.160.10]:

FIGURE F.6
INSTRUCTIONS:
You can:
   1) Enter the appropriate information for your system
      and press ENTER or
   2) Press ENTER if you are satisfied with the value
      within the brackets [ ] and want to go on to the
      next parameter or
   3) Press ESC if you want to exit.

NETMASK - A string of 0's and 1's that mask or screen out
the host part of an IP address so that only the network
part of the address remains.

Netmask[255.255.255.0]:

FIGURE F.7
Your current configuration parameters are:

Hostname: TSx000
System IP: 192.168.160.10
Domain Name: mycompany.com
DNS: 127.0.0.1
Gateway: 192.168.160.10
Mask: 255.255.255.0

Are all these parameters correct (Y)es or (N)o [N] :

**FIGURE F.8**

Type ‘y’ if all parameters are correct. Type ‘n’ or just press ENTER if not all the parameters are correct and you want to go back and redo them.

If ‘n’ is entered, this is displayed:

Type 'c' to go back and CORRECT the current configuration parameters or 'q' to QUIT:

Type ‘c’ to go back and CORRECT the current configuration parameters or ‘q’ to QUIT.
If ‘y’ is entered, Figure F.9 is displayed. This figure explains what saving to flash means. Type ‘y’ if you want to save to flash. Type ‘n’ if you don’t want to save to flash. You can now continue TS configuration using the web browser by typing in the IP address of the TS. If you choose to not save to flash, all the new configuration will be lost if you were to reboot the TS. However, all configuration will be kept if you saved to flash.

Figure F.9

NOTE: Using telnet to configure the Cyclades-TS, if you reconfigure the IP address of the Ethernet interface you are supposed to have your telnet connection lost. In that case, close the telnet client and reopen it using this time the new IP address configurated for that TS box.
Using Wizard through WEB

The web interface supports wizards for the serial ports configuration. The following profiles are supported for the serial ports:

- Console Access Server (CAS) profile
- Terminal Server (TS) profile
- Remote Access Server (RAS) profile
- Automation Profile (a subset of CAS profile, only for the Cyclades-TS100)

Most of the applications should fit in one of these profiles, so the wizard is a useful tool to ease the configuration of the serial ports. The web interface will access the wizard files for each profile:

- `/etc/portslave/pslave.wiz.cas` (CAS profile)
- `/etc/portslave/pslave.wiz.ts` (TS profile)
- `/etc/portslave/pslave.wiz.ras` (RAS profile)
- `/etc/portslave/pslave.wiz.auto` (Automation profile)

The wizard configuration is set by pressing one of the buttons which appear in the Wizard section in the Serial Port Configuration page. The default parameters will then be set in the page and, after that, the user must change the parameters which need to be changed, and then press the Submit button.

The tables below show the values of the parameters which will be set for each profile. The parameters in bold are the ones whose value will probably be required to change.
## Console Access Server (CAS) profile:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed</td>
<td>9600</td>
<td>Port speed - 9600bps</td>
</tr>
<tr>
<td>datasize</td>
<td>8</td>
<td>8 bits</td>
</tr>
<tr>
<td>stopbits</td>
<td>1</td>
<td>1 stop bit</td>
</tr>
<tr>
<td>parity</td>
<td>none</td>
<td>no parity</td>
</tr>
<tr>
<td>flow</td>
<td>hard</td>
<td>hardware flow control</td>
</tr>
<tr>
<td>dcd</td>
<td>0</td>
<td>not sensitive to DCD signal</td>
</tr>
<tr>
<td>sysutmp</td>
<td>1</td>
<td>write the users in utmp log file</td>
</tr>
<tr>
<td>syswtmp</td>
<td>0</td>
<td>Do not write the users in wtmp log file.</td>
</tr>
<tr>
<td>authtype</td>
<td>radius</td>
<td>Authentication can be through Radius, TACACS+ or Local.</td>
</tr>
<tr>
<td>authhost1</td>
<td>200.200.200.2</td>
<td>change it to the authentication server of your environment</td>
</tr>
<tr>
<td>accthost1</td>
<td>200.200.200.2</td>
<td>change it to the accounting server of your environment</td>
</tr>
<tr>
<td>radtimeout</td>
<td>3</td>
<td>3 minutes timeou</td>
</tr>
<tr>
<td>secret</td>
<td>cyclades</td>
<td>change it to the secret of the RADIUS/ TACACS+ server</td>
</tr>
<tr>
<td>radretries</td>
<td>5</td>
<td>5 retries before giving up authenticating</td>
</tr>
<tr>
<td>rradnullpass</td>
<td>0</td>
<td>Don't allow users with null passwords.</td>
</tr>
<tr>
<td>protocol</td>
<td>socket_server</td>
<td>Telnet protocol. SSH (socket_ssh) and raw (raw_data) are also supported.</td>
</tr>
<tr>
<td>ipno</td>
<td>192.168.1.101+</td>
<td>This value can be kept unless you access through IP address is required.</td>
</tr>
<tr>
<td>socket_port</td>
<td>7001+</td>
<td>Change it to the TCP port to be used by the first serial port; keep the Incremented option on.</td>
</tr>
<tr>
<td>issue</td>
<td>\nWelcome to...</td>
<td>This will be the banner when a login is required.</td>
</tr>
<tr>
<td>prompt</td>
<td>%h login</td>
<td>Login prompt. %h is the hostname.</td>
</tr>
<tr>
<td>term</td>
<td>vt100</td>
<td>Other terminal types are available (ansi, linux).</td>
</tr>
<tr>
<td>tx_interval</td>
<td>100</td>
<td>Send buffer received to the application each 100ms.</td>
</tr>
<tr>
<td>poll_interval</td>
<td>0</td>
<td>Don't send modem state commands.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Comments</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------</td>
<td>-----------------------------------------------------------</td>
</tr>
<tr>
<td>idletimeout</td>
<td>0</td>
<td>Don’t finish the session by idle timeout.</td>
</tr>
<tr>
<td><strong>data_buffering</strong></td>
<td>0</td>
<td><strong>Data buffering disabled.</strong></td>
</tr>
<tr>
<td>DB_timestamp</td>
<td>0</td>
<td>Don’t include time in the data buffering.</td>
</tr>
<tr>
<td>alarm</td>
<td>0</td>
<td>Don’t generate alarm syslogs.</td>
</tr>
<tr>
<td>syslog_buffering</td>
<td>0</td>
<td>Don’t generate syslogs for data buffering.</td>
</tr>
<tr>
<td>dont_show_DB_menu</td>
<td>1</td>
<td>Don’t show DB menu when the session is opened.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>break_sequence &quot;break. this sequence will generate a BREAK in the serial port, in an SSH session.</td>
</tr>
<tr>
<td>sniff_mode</td>
<td>out</td>
<td>Only output packets can be traced.</td>
</tr>
<tr>
<td>admin_users</td>
<td>peter, john</td>
<td>If the sniff must be disabled and these users don’t exist, this value can be kept.</td>
</tr>
<tr>
<td>multiple_sessions</td>
<td>no</td>
<td>Don’t accept multiple sessions for sniff.</td>
</tr>
<tr>
<td>escape_char</td>
<td>^z</td>
<td>This character will cause a sniff session to switch to the command mode.</td>
</tr>
<tr>
<td>serverfarm</td>
<td>Server_connected serial&lt;port_number&gt;</td>
<td>Name of the SSH server connected to the serial port</td>
</tr>
</tbody>
</table>

**TS profile:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed</td>
<td>9600</td>
<td>Port speed - 9600 bps.</td>
</tr>
<tr>
<td>datasize</td>
<td>8</td>
<td>8 bits.</td>
</tr>
<tr>
<td>stopbits</td>
<td>1</td>
<td>1 stop bit.</td>
</tr>
<tr>
<td>parity</td>
<td>None</td>
<td>No parity</td>
</tr>
<tr>
<td>flow</td>
<td>hard</td>
<td>Hardware flow control.</td>
</tr>
<tr>
<td>dcd</td>
<td>1</td>
<td>Sensitive to DCD signal.</td>
</tr>
<tr>
<td>sysutmp</td>
<td>1</td>
<td>Write the users in utmp log file.</td>
</tr>
<tr>
<td>syswtmp</td>
<td>0</td>
<td>Don't write the users in wtmp log file.</td>
</tr>
</tbody>
</table>
### Parameter | Value | Comments
---|---|---
authtype | none | No authentication; the next six parameters will be used only if authentication has radius or tacacs+.
authhost1 | 200.200.200.2 | Change it to the authentication server of your network
accthost1 | 200.200.200.2 | Change it to the accounting server of your network
radtimeout | 3 | 3 minutes of timeout.
secret | cyclades | Change it to the secret of your RADIUS server.
radretries | 5 | 5 retries before giving up authenticating.
radnullpass | 0 | Don't allow users with null passwords.
protocol | telnet | The other protocols are login, rlogin, ssh and socket_client.
socket_port | 23 | TCP port (22 for ssh, 513 for rlogin).
host | 200.200.200.3 | Change it to the server to which the TS will log in.
issue | \r\nWelcome to | This will be the banner when a login is required.
prompt | %h login | Login prompt. %h is the hostname.
term | vt100 | Other terminal types are available (ansi, linux).

### RAS profile:

| Parameter | Value | Comments |
---|---|---|
speed | 57600 | This speed is normally used for communication with modems. |
datasize | 8 | 8 bits character. |
stopbits | 1 | 1 stop bit. |
parity | none | No parity. |
flow | hard | Hardware flow control. |
dcd | 1 | Sensitive to DCD signal. |
sysutmp | 1 | Write the users in utmp log file. |
syswtmp | 0 | Don't write the users in wtmp log file. |
authtype | radius | RADIUS authentication; the next six parameters will be used only if authentication has radius or tacacs+. |
### Parameter | Value | Comments
---|---|---
authhost1 | 200.200.200.2 | Change it to the authentication server of your network.
accthost1 | 200.200.200.2 | Change it to the accounting server of your network.
radtimeout | 3 | 3 minutes of timeout.
secret | cyclades | Change it to the secret of the RADIUS/. TACACS+ server.
raddrtries | 5 | 5 retries before giving up authenticating.
radnullpass | 0 | Don't allow users with null passwords.
protocol | ppp | The other protocols are slip and cslip.
ipno | 200.200.200.11+ | Change it to the IP address of the remote user connected to the first port. Keep the Incremented option on.
issue | 

Welcome to ... | This will be the banner when a login is required.
prompt | %h login | Login prompt. %h is the hostname.
etmask | 255.255.255.255 | This option are to be changed only for LAN-to-LAN profile.
mtu | 1500 | This is the default MTU
mru | 1500 | This is the default MRU
initchat | TIMEOUT 10 | This chat script fits for most of the modems
autopp | %i:%j novj ... | Van Jacobson disabled, ACCM with characters XON/XOFF, IPX disabled, CCP disabled, authentication PAP, callback enabled.
pppopt | %i:%j novj ... | Van Jacobson disabled, ACCM with characters XON/XOFF, IPX disabled, CCP disabled.

### Automation profile:

| Parameter | Value | Comments |
---|---|---|
speed | 9600 | Port speed - 9600bps.|
datasize | 8 | 8 bits character.|
stopbits | 1 | 1 stop bit.|
parity | none | No parity.|

Appendix F  Configuration Wizard
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>flow</td>
<td>hard</td>
<td>Hardware flow control.</td>
</tr>
<tr>
<td>dcd</td>
<td>0</td>
<td>Not sensitive to DCD signal.</td>
</tr>
<tr>
<td>media</td>
<td>rs232</td>
<td>Change this option if RS485 media is used.</td>
</tr>
<tr>
<td>modbus_smode</td>
<td>ascii</td>
<td>Change this option if RTU serial mode is used.</td>
</tr>
<tr>
<td>sysutmp</td>
<td>1</td>
<td>Write the users in utmp log file.</td>
</tr>
<tr>
<td>syswttmp</td>
<td>0</td>
<td>Don't write the users in wtmp log file.</td>
</tr>
<tr>
<td>authtype</td>
<td>none</td>
<td>No authentication.</td>
</tr>
<tr>
<td>protocol</td>
<td>modbus</td>
<td>MODBUS protocol.</td>
</tr>
<tr>
<td>socket_port</td>
<td>520</td>
<td>MODBUS port.</td>
</tr>
</tbody>
</table>
APPENDIX G GENERATING ALARM AND SYSLOG

Versions 1.3.3 and later

This appendix shows the characteristics of the Alarm for Data Buffering that is implemented for all the TSxk family. It is divided in five parts:

1. Syslog-ng and its configuration
2. Alarm, sendmail, sendsms and snmtrap
3. Example of the configuration to use syslog_buffering
4. Example of the configuration to use alarm feature
5. Example of the configuration to use multiples syslog servers.

1. Syslog-ng

The syslog-ng reads from sources (files, TCP/UDP connections, syslogd clients), filters the messages and takes an action (writes in files, sends snmptrap, pager, e-mail or syslogs).

The configuration file is read at startup and is reread after receipt of a hangup (HUP) signal. When reloading the configuration file, all destination files are closed and reopened as appropriate.

You will need to define sources, filters and actions (destinations), and after you’ll connect them as explained below.

You can specify several global options to syslog-ng in the options statement:

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

where optn can be any of the following:
- `time_reopen(n)`: the time to wait before a died connection is reestablished.
- `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 attach.
- `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 creating no-existing 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

To define sources use this statement:

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

where:

- `identifier`: it has to uniquely identify this given source;
- `source-driver`: it is a method of getting a given message.
- `params`: each source-driver may take parameters, some of them required, some of them optional.
The following source-drivers are available:

a) **internal()**
   - messages generated internally in syslog-ng

b) **unix_stream(filename [options]) and unix_dgram(filename [options])**
   - they open the given AF_UNIX socket, and start listening on them 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,
       - can be used only with unix_stream. Default: yes
       - max-connections(n) - limits the number of simultaneously opened connections. Can be used
         only with unix_stream. Default: 10.

c) **tcp([options]) and udp([options])**
   - these drivers let you receive messages from the network, and as the name of the drivers show, you can
     use both TCP and UDP.
   - none of tcp() and udp() drivers require positional parameters. By default the bind to 0.0.0.0:514, which
     means that syslog-ng will listen on all available interfaces.
   - options: ip(<ip address>) - the IP address to bind to. 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.

d) **file(filename)**
   - it opens the specified file, and reads messages.

e) **pipe(filename)**
   - it opens a named pipe with the specified name, and listens for messages. (you'll need to create the pipe using mkfifo command).
Some examples:

1. To read from a file: source <identifier> {file(filename);};
   Example to read messages from “/temp/file1” file:
   source file1 {file("/temp/file1");};
   Example to receive messages from kernel:
   source s_kernel { file("/proc/kmsg");};

2. To receive messages from local syslogd clients:
   source sysl {unix-stream("/dev/log");};

3. To receive messages from remote syslogd clients:
   source s_udp { udp(ip(<cliente ip>) port(<udp port>)); };  
   Example to listen messages from all machines on UDP port 514:
   source s_udp { udp(ip(0.0.0.0) port(514));};
   Example to listen messages from one client (IP address=10.0.0.1) on UDP port 999:
   source s_udp_10 { udp(ip(10.0.0.1) port(999));};

To define filters use this statement:

   filter <identifier> { expression; };  

   where: identifier - has to uniquely identify this given filter
   expression - boolean expression using internal functions, which has to evaluate to true for the message to pass.
The following internals functions are available:

a) facility(<facility code>):  
   - selects messages based on their facility code.

b) level(<level code>) or priority(<level code>):  
   - selects messages based on their priority

c) program(<string>):  
   - tries to match the <string> to the program name field of the log message

d) host(<string>):  
   - tries to match the <string> to the hostname field of the log message

e) match(<string>):  
   - tries to match the <string> to the message itself.

Some examples:

1. To filter by facility: filter f_facility { facility(<facility name>); };
   Examples:
   filter f_daemon { facility(daemon); };
   filter f_kern { facility(kern); };
   filter f_debug { not facility(auth, authpriv, news, mail); };

2. 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); }

3. To filter by matching one string in the received message: filter f_match { match("string"); }
   Example to filter by matching the string "named":
   filter f_named { match("named"); }

4. To filter ALARM messages:
   filter f_alarm { facility(local[0+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(local1) and level(info) and match("ALARM") and match("kernel panic"); }
   Example to filter ALARM message with the string “root login”:
   filter f_root { facility(local1) and level(info) and match ("ALARM") and match("root login"); }

5. Example the filter to eliminate sshd debug messages
   filter f_sshd_debug { not program("sshd") or not level(debug); }

6. To filter the syslog_buffering:
   filter f_syslog_buf { facility(local[0+<conf.DB_facility>]); }

To define actions use this statement:

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

where: identifier - has to uniquely identify this given destination.
   destination-driver: it is a method of outputing a given message.
   params: each destination-driver may take parameters, some of them required, some of them optional.

The following destination drivers are available:

a) 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 where the message is originated from.
  - FACILITY - the name of the facility, the message is tagged as coming from.
  - 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 - it equal 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:
  - logfifo_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.
  - encrypt(yes/no) - encrypt the resulting file.
  - compress(yes/no) - compress the resulting file using zlib.
  - owner(name), group(name), perm(mask) - equal global options
  - template(“string”) - syslog-ng write the “string” in the file.

b) pipe(filename [options])
- this driver sends messages to a named pipe.
- available options:
  - owner(name), group(name), perm(mask) - equal global options
• template("string") - syslog-ng write the “string” in the file. You can use the MACROS in the string.

c) unix-stream(filename) and unix-dgram(filename)
  - this driver sends messages to a unix socket en either SOCKET_STREAM or SOCK_DGRAM mode.

d) 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.

e) usertty(<username>)
  - this driver writes messages to the terminal of a logged-in username.

f) program(<program name and arguments>)
  - this driver fork()’s executes the given program with the arguments and sends messages down to the stdin of the child.

Some examples:

1. To send e-mail:

   destination <ident> { pipe("/dev/cyc_alarm" template("sendmail <pars>")));;
   where ident: uniquely identify this destination
   pars: -t <name>[,.<name>]: To address
         [-c <name>[,.<name>]]: CC address
         [-b <name>[,.<name>]]: Bcc address
         [-r <name>[,.<name>]]: Reply-to address
         -f <name>: From address
         -s "<text>"": Subject
         -m "<text message>"": Message
         -h <IP address or name>: SMTP server
         [-p <port>]: port used. default: 25

   To mount the message, use this macros:
$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

Example to send e-mail to z@none.com (SMTP’s IP address 10.0.0.2) from the e-mail address a@none.com with subject “TSxK-ALARM”. The message will carry the current date, the hostname of this TS and the message that was received from the source:

destination d_mail1 {
    pipe("/dev/cyc_alarm"
         template("sendmail -t z@none.com -f a@none.com -s "TSxK-ALARM" -m "$FULLDATE $HOST $MSG" -h 10.0.0.2"));
}

2. To send to pager server (sms server):

destination <ident> {pipe("/dev/cyc_alarm" template("sendsms <pars>"));

where ident: uniquely identify this destination
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>

Example to send a pager to phone number 123 (Pager server at 10.0.0.1) with message carrying the current date, the hostname of this TS and the message that was received from the source:

destination d_pager {
    pipe("/dev/cyc_alarm"
template("sendsms -d 123 -m "$FULLDATE $HOST $MSG" 10.0.0.1");

3. To send snmptrap:
   destination <ident> {pipe("/dev/cyc_alarm" template("snmptrap <pars>"))); 
where ident: uniquely identify this destination
pars: -v 1

   <snmptrapd IP address>
   public : community
   ": enterprise-oid
   ": agent/hostname
   <trap number> : 2-Link Down, 3-Link Up, 4-Authentication Failure
   0 : 
   ": host-uptime
   .1.3.6.1.2.1.2.2.1.2.1 s : string
   "$<message - max. size 250 characters>"

Example to send a Link Down trap to server at 10.0.0.1 with message carrying the current date, the hostname of this TS and the message that received from the source:

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

4. To write in file:
   destination d_file { file(<filename>);};
Example send message to console:
   destination d_console { file("/dev/ttyS0");};
Example write message in /var/log/messages file:
   destination d_message { file("/var/log/messages"); }; 

5. To write messages to the session of logged-in user:
destination d_user { usertty("<username>"); }; 
Example to send message to all sessions with root user logged:
destination d_userroot { usertty("root"); }; 

6. To send message to remote syslogd server:
destination d_udp { udp("<remote IP address>" port(514)); }; 
Example to send syslogs to syslogd located at 10.0.0.1:
destination d_udp1 { udp("10.0.0.1" port(514)); }; 

To connect the sources, filters and actions (any message coming from one of the listed sources, matching the filters (each of them) is sent to the listed destinations). Use this statement:

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

where: Sx - identifier of the sources defined before  
Fx - identifier of the filters defined before  
Dx - identifier of the actions/destinations defined before  

Examples:

1. To send all messages received from local syslog clients to console
   log { source(sysl); destination(d_console);};

2. To send only messages with level alert and received from local syslog clients to all logged root user:
   log { source(sysl); filter(f_alert); destination(d_userroot); }; 

3. To writes all messages with levels info, notice or warning and received from syslog clients (local and remotes) to /var/log/messages file:
log { source(sysl); source(s_udp); filter(f_messages); destination(d_messages); }

4. To send e-mail if message received from local syslog client has the string “kernel panic”:
   log { source(sysl); filter(f_kpanic); destination(d_mail1); }

5. To send e-mail 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); }

6. To send messages with facility kernel and received from syslog clients (local and remote) to remote syslogd:
   log { source(sysl); source(s_udp); filter(f_kern); destination(d-udp1); }

2. Alarm, Sendmail, Sendsms and Snmptrap

2.1. Alarm

This feature is available only in the Console Server Application.

TS sends messages using pager, e-mail or snmptrap if the serial port receives message with specific string.

To configure this feature you will need:

1. to activate alarm in Portslave configuration file (parameter all.alarm - 0 inactive or <> 0 active)

2. to configure filters in syslog-ng configuration file:
   filter f_alarm { facility(local[0+DB_facility]) and level(info) and match("ALARM") and match("<your string>"); };

   For example, to filter ALARM message with the string “kernel panic”:
   filter f_kpanic { facility(local1) and level(info) and match("ALARM") and match("kernel panic"); };

For example, to filter ALARM message with the string “root login”:

```
filter f_root { facility(local1) and level(info) and match("ALARM") and match("root login"); };
```

3. to configure actions in syslog-ng configuration file.

Examples (see more details in syslog-ng examples):
- to send e-mail: destination d_mail { pipe("/dev/cyc_alarm" template("sendmail <pars>")));;
- to send pager: destination d_pager {pipe("/dev/cyc_alarm" template("sendsms <pars>")));;
- to send snmptrap: destination d_trap {pipe("/dev/cyc_alarm" template("snmptrap <pars>")));;

4. to connect filters and actions in syslog-ng configuration file.

Example: alarm is active and if the serial port receives the string “kernel panic”, one message will be sent to the pager.

```
log (source(syl); filter(f_kpanic); destination(d_trap); destination(d_pager); };
```

### 2.2. Sendmail

Sendmail sends a message to a SMTP server. It is not intended as a user interface routine; it is used only to send pre formatted messages. Sendmail reads all parameters in command line.

If the SMTP server does not answer the SMTP protocol requests sent by sendmail, the message is dropped.

**Synopsis:** `sendmail -t <name>[,<name>][ -c <name> [,<name>]] [-b <name> [,<name>]] [-r <name>] -f <name> -s <text> -m <text> -h <SMTP server> [-p <smtp-port>]

where:
- `-t <name>[,<name>]`
“To: “. Required. Multi-part allowed (multiple names are separated by commas). Names are expanded as explained below.

[-c <name> [,<name>]]
“Cc: “. Optional. Multi-part allowed (multiple names are separated by commas).

[-b <name> [,<name>]]
“Bcc:”. Optional. Multi-part allowed (multiple names are separated by commas).

[-r <name> ]
“Reply-To: “. Optional. Use the Reply-To: field to make sure the destination user can send a reply to a regular mailbox.

-f <name>
“From: “ Required.

-s <text>
“Subject: “. Required.

-m <text>
“body”. The message body.

-h <SMTP server>
Required. IP address or name of the SMTP server.

[-p <SMTP port>
Optional. The port number used in the connection with the server. Default: 25.

<name>: Any email address.
For example, to send e-mail to z@none.com (SMTP’s IP address 10.0.0.2) from the e-mail address a@none.com with subject “TS sendmail test “.

```
sendmail -t z@none.com -f a@none.com -s “TS sendmail test” -m “Sendmail test. \n Is it OK??? “ -h 10.0.0.2
```

### 2.3. Sendsms

The sendsms is the Linux command line client for the SMSLink project (Philippe Andersson - “Les Ateliers du Heron”). It accepts command line parameters that define the message to be sent, and transmits them to the SMS server process running on the designated server. The sendsms was developed specifically for easy calling from shell scripts or similar situations.

**Synopsis:**

```
sendsms [-r] [-g] [-v] -d dest (-m message or -f msgfile) [-u user] [-p port] server, where:
```

- `-r` : Reporting. Additional info will be included in the message printed on stderr (namely, the device name used by the server to send the SMS out, and the message ID attributed to the SMS by the module’s SIM card). If any of these items is missing or can’t be parsed, a value of “??” will be returned.

- `-g` : Turns debugging on. Will output the entire dialog with the server on stderr (and more).

- `-h` : Displays a short help message and exits.

- `-v` : Displays version information and exits.
-d dest: Required. The GSM network address (i.e. phone number) of the mobile phone the message is to be sent to. Supported format is: [int. prefix - country code] area code - phone number. The international prefix can be either “+” or “00” (or any other value supported by the GSM network provider the server is subscribed to). Some separation characters can be used to beautify the number, but they are purely cosmetic and will be stripped by the server. Those characters are [./- ]. The pause character (`,`) is not supported. Regarding the international country code, don’t forget that its necessity is to be considered respective to the SMS gateway location (the host this client program is connecting to), not the location where the client is run from. In case of doubt, please contact the SMS server administrator for your network. Please always include the area code (even when sending to a destination in the same “area”, i.e. on the same network). The number without the area code, though syntactically correct and accepted by the network, would never get delivered (at least, that’s my experience with Proximus — YMMV).

-m message : Required (Use one and only one of “-m” or “-f”). The text of the message to be sent. Unless made up of a single word, it will have to be quoted for obvious reasons. Maximum length is 160 characters. A longer message will be truncated (the user will be warned about it), but the message will still be sent. At the present time, only 7bit ASCII is supported for the message text.

-f msgfile : Required (use one and only one of “-m” or “-f”). The name of a text file where the message to send is to be read from. This file can contain multiple lines of text (they will be concatenated), but its total length can’t exceed 160 characters. A longer text will be truncated (the user will be warned about it), but the message will still be sent. The special file ‘-‘ means that input will be read from stdin. At the present time, only 7bit ASCII is supported for the message text.

-u user : Optional. The server module requires the user to identify himself for logging purposes. No authentication is performed on this info though. If this parameter is omitted, sendsms will send the Unix username of the current user. This parameter allows you to override this default behavior (might be useful in the case of automated sending).

-p port : Optional. Communication port on the target server. If provided here, this value will be used to connect to the server. If omitted, the client will query the local system for the port number associated with the
“well known service” sms (as defined in /etc/services). If that doesn’t return an answer, the compiled-in default value 6701 will be used.

    server   : Required. The host name or IP address of the computer where the SMS gateway server process is running. By default, this server will be listening on TCP port 6701.

Upon success (when the server module reports that the message was successfully sent), sendsms returns 0. When a problem occurs, a non zero value is returned. Different return values indicate different problems. A return value of 1 indicates a general failure of the client program.

COPYRIGHT: SMSLink is (c) Les Ateliers du Heron, 1998 by Philippe Andersson <philipa@tiscalinet.be>. It has been originally written for Scitex Europe, S.A. Part of the code is (c) Riccardo Facchetti. The code also includes contributions from Philipp Klaus <pklaus@access.ch> and numerous others. All contributors are acknowledged in the CHANGEOLOG document, and in the comment headers of the source files they modified. SMSLink has been released to the public under the GNU GPL.

Example to send a pager to phone number 123 (Pager server at 10.0.0.1) with message:

    sendsms -d 123 -m “Hi, it is a test message send from TSxK using sendsms” 10.0.0.1

2.4. Snmptrap

Snmptrap is an SNMP application that uses the TRAP-PDU Request to send information to a network manager. One or more fully qualified object identifiers can be given as arguments on the command line. A type and a value must accompany each object identifier. Each variable name is given in the format specified. If any of the required version 1 parameters: enterprise-oid, agent and uptime are specified as empty, it defaults to “.1.3.6.1.4.1.3.1.1”, hostname and host-upptime respectively.

Synopsis:
snmptrap -v 1 [-Ci] [common arguments] enterprise-oid agent generic-trap specific-trap uptime [objectID type value]...

snmptrap -v [2|3] [-Ci] [common arguments] uptime trap-oid [objectID type value]...

where:

-Ci : Optional. It sends INFORM-PDU

common arguments: required. They are: SNMP server IP address and community.

enterprise-oid: required, but it can be empty ("").

agent: required, but it can be empty(""). The agent name.

generic-trap: required. The generic trap number: 2 (link down), 3 (link up), 4 (authentication failure), ...

specific-trap: required. The specific trap number.

uptime: required.

[objectID type value]: optional. objectID is the object oid, you want to inform its value to server.

If the network entity has an error processing the request packet, an error packet will be returned and a message will be shown, helping to pinpoint in what way the request was malformed. If there were other variables in the request, the request will be resent without the bad variable.

For example, to send a Link Down trap to server at 10.0.0.1 with interfaces.iftable.ifentry.ifdescr :

snmptrap -v 1 10.0.0.1 public "" 2 0 "" .1.3.6.1.2.1.2.2.1.2.1 s ""TSxK: serial port number 1 is down""
3. Syslog-ng configuration to use with syslog buffering feature

This configuration example is to use syslog buffering feature, and to send the messages to remote syslogd (10.0.0.1).

In the pslave.conf file the parameters of the syslog buffering feature are configured as:

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

The syslog-ng.conf file need these lines:

```
# 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); ];
```

4. Syslog-ng configuration to use with alarm feature

This configuration example is to use alarm feature.

In the pslave.conf file the parameters of the alarm feature are configured as:

```
all.alarm         1
conf.DB_facility         2
```
The syslog-ng.conf file need these lines:

```plaintext
# local syslog clients
source src { unix-stream("/dev/log"); }

# To filter ALARM message with the string "kernel panic":
filter f_kpanic { facility(local2) and level(info) and match("ALARM") and match("kernel panic"); }

# To filter ALARM message with the string "root login":
filter f_root { facility(local1) and level(info) and match("ALARM") and match("root login"); }

# To send e-mail to z@none.com (SMTP’s IP address 10.0.0.2)
# from the e-mail address a@none.com with subject “TSxK-ALARM”.
# The message will carry the current date, the hostname
# of this TS and the message that was received from the source.
destination d_mail1 {
    pipe("/dev/cyc_alarm"
        template("sendmail -t z@none.com -f a@none.com -s "TSxK-ALARM\" \ -m "$FULLDATE $HOST $MSG\" -h 10.0.0.2");
    );
}

# Example to send a pager to phone number 123 (Pager server at 10.0.0.1) with message
# carrying the current date, the hostname of this TS and the message that was received from the source:
destination d_pager {
    pipe("/dev/cyc_alarm"
        template("sendsms -d 123 -m "$FULLDATE $HOST $MSG\" 10.0.0.1");
    );
}
```
# Example to send a Link Down trap to server at 10.0.0.1 with message carrying the current
date, the hostname of this TS and the message that received from the source:
destination d_trap {
    pipe("/dev/cyc_alarm"
        template("snmptrap -v 1 10.0.0.1 public "}\" "}\" 2 0 "}\" 
        .1.3.6.1.2.1.2.2.1.2.1 s "}\"FULLDATE $HOST $MSG", ");
};

# To send e-mail and snmptrap if message received from local syslog client has the string “kernel panic”:
log { source(sysl); filter(f_kpanic); destination(d_mail1); destination(d_trap); };

# To send e-mail 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); };

5. Syslog-ng configuration to use with multiple remote syslog servers

This configuration example is to use multiple remote syslog servers.

In the pslave.conf file the facility parameter is configured as:
    conf.facility 1

The syslog-ng.conf file need these lines:

    # local syslog clients
    source src { unix-stream("/dev/log"); };
# remote server 1 - IP address 10.0.0.1 port default
destination d_udp1 { udp("10.0.0.1");};

# remote server 2 - IP address 10.0.0.2 port 1999
destination d_udp2 { udp("10.0.0.2" port(1999));};

# filter messages from facility local1 and level info to warning
filter f_local1 { facility(local1) and level(info..warn)};

# filter messages from facility local 1 and level err to alert
filter f_critic { facility(local1) and level(err .. alert)};

# send info, notice and warning messages to remote server udp1
log { source(src); filter(f_local1); destination(d_udp1); };

# send error, critical and alert messages to remote server udp2
log { source(src); filter(f_critic); destination(d_udp2); };
APPENDIX H  CERTIFICATE FOR HTTP SECURITY

Obtaining a Signed Digital Certificate

A certificate for the HTTP security is created by a CA (Certification Authority). The most usual procedure to obtain a certificate is:

- Generation of the public and private keys, using a public key algorithm like RSA or X509. The keys can be generated by using a key generator software. In a Linux computer, this can be done using the OpenSSL package, through the following command:

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

If this command is used, the following information is required:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country Name (2 letters code) [AU]:</td>
<td>The country code with two letters.</td>
</tr>
<tr>
<td>State or Province Name (full name) [Some-State]:</td>
<td>Provide the full name (not the code) of the state.</td>
</tr>
<tr>
<td>Locality Name (e.g., city) []:</td>
<td>Enter the name of your city.</td>
</tr>
<tr>
<td>Organization Name (e.g., company) [Internet Widgits Pty Ltd]:</td>
<td>Organization that you work or want to certificate for.</td>
</tr>
<tr>
<td>Organizational Unit Name (e.g., section) []:</td>
<td>Department or section which you work.</td>
</tr>
<tr>
<td>Common Name (e.g., your name or your server's hostname) []:</td>
<td>Name of the machine where the certificate must be installed.</td>
</tr>
<tr>
<td>Email Address []:</td>
<td>Your email address or the administrator email address.</td>
</tr>
</tbody>
</table>
The other requested information can be skipped.

- The certificate signing request (CSR) generated by the command above contains some personal (or corporate) information and its public key. The next step is to submit the CSR and some personal data to the CA. This service can be requested by accessing the CA website and is not free, and there is a list of CA's in the URL [http://www.pki-page.org/](http://www.pki-page.org/).

- The request will be analyzed by the CA, for policy approval and to be signed.

- After the approval, the CA will send a certificate file to the origin, which we will call Cert.cer, for example purposes. The certificate is also stored on a directory server.

- The certificate must be installed in the GoAhead web server, by following these instructions:

  1. Open a Cyclades Terminal Server session and do the login.

  2. Load the files Cert.cer (certificate file received from the CA) and private.key (private key generated by openssl) to a temporary directory (/tmp), using FTP or NFS service.

  3. Join the certificate with the private key into the file
     /web/server.pem.
     
     #cat Cert.cer private.key > /web/server.pem

  4. Copy the certificate to the file
     /web/cert.pem
     
     #cp Cert.cer /web/cert.pem

  5. Include the files /web/server.pem and /web/cert.pem in /etc/config_files.
6. Save the configuration in flash.
   #saveconf

7. The certification will be effective in the next reboot.
APPENDIX I USING MODBUS PROTOCOL IN CAS PROFILE

MODBUS is an application layer messaging protocol for client/server communication which is widely used in the industrial automation. It is a confirmed service protocol and offers many services specified by function codes, like reading and writing registers on PLCs.

A protocol converter for the MODBUS protocol over the TCP/IP communication stack (Modbus/TCP) is implemented in Cyclades-TS and converts Modbus/TCP ADUs from the Ethernet interface to plain MODBUS message frames over a serial RS-232 or RS-485 interface, and vice-versa, supporting both serial modes (ASCII and RTU).

FIGURE I.1 - MODBUS APPLICATION

Cyclades-TS Installation & Service Manual

Appendix I Using MODBUS Protocol in CAS Profile
In this example, the Automation Application running in the Workstation (local or remote) controls the PLCs connected to the serial port (RS-485) of the Cyclades-TS100 using MODBUS/TCP protocol. The connection is opened using Cyclades-TS100 Ethernet IP address and TCP port = 502. Cyclades-TS100 accepts the incoming connection and converts MODBUS/TCP ADUs (packets) to plain MODBUS frames and sends them over the serial port. On the other hand, the MODBUS frames received from the serial port are converted to MODBUS/TCP ADUs and sent through the TCP connection to the Automation Application.

The configuration described earlier for Console Access Servers (see Chapter 6) should be followed with the following exceptions for this example:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>all.authtype</td>
<td>There are several authentication type options: local (authentication is performed using the /etc/passwd file), radius (authentication is performed using a Radius authentication server), TacacsPlus (authentication is performed using a TacacsPlus authentication server), none, local/radius (authentication is performed locally first, switching to Radius if unsuccessful), radius/local (the opposite of the previous option), RadiusDownLocal (local authentication is tried only when the Radius server is down), local/TacacsPlus (authentication is performed locally first, switching to TacacsPlus if unsuccessful), TacacsPlus/local (the opposite of the previous option), TacacsPlusDownLocal (local authentication is tried only when the TacacsPlus server is down). Note that this parameter controls the authentication required by the Cyclades-TS. The authentication required by the device to which the user is connecting is controlled separately.</td>
<td>none</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value for This Example</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>all.protocol</td>
<td>For the console server profile, the possible protocols are socket_server (when telnet is used), socket_ssh (when ssh version one or two is used), raw_data (to exchange data in transparent mode – similar to socket_server mode, but without telnet negotiation, breaks to serial ports, etc.), or modbus (an application layer messaging protocol for client/server communication widely used for industrial automation).</td>
<td>modbus</td>
</tr>
<tr>
<td>all.modbus_smode</td>
<td>Communication mode through the serial ports. This parameter is meaningful only when modbus protocol is configured. The valid options are ascii (normal TX/RX mode) and rtu (some time constraints are observed between characters while transmitting a frame). If not configured, ASCII mode will be assumed.</td>
<td>ascii</td>
</tr>
</tbody>
</table>

**FIGURE I.2 - MODBUS PSLAVE.CONF PORT SPECIFIC PARAMETERS (CONT.)**

*(ONLY WHERE IT DIFFERS FROM THE STANDARD CAS PROFILE)*

Note: The MODBUS port can be configured in the file /etc/services, changing the corresponding line. By default, the port is 502, as specified in Modbus/TCP draft to the IETF.
APPENDIX J  LINUX-PAM

Overview

Linux-PAM (Pluggable Authentication Modules for Linux) is a suite of shared libraries that enable the local system administrator to choose how applications authenticate users.

In other words, without (rewriting and) recompiling a PAM-aware application, it is possible to switch between the authentication mechanism(s) it uses. Indeed, one may entirely upgrade the local authentication system without touching the applications themselves.

It is the purpose of the Linux-PAM project to separate the development of privilege granting software from the development of secure and appropriate authentication schemes. This is accomplished by providing a library of functions that an application may use to request that a user be authenticated. This PAM library is configured locally with a system file, /etc/pam.conf (or a series of configuration files located in /etc/pam.d/) to authenticate a user request via the locally available authentication modules. The modules themselves will usually be located in the directory /lib/security and take the form of dynamically loadable object files.

The Linux-PAM authentication mechanism gives to the system administrator the freedom to stipulate which authentication scheme is to be used. He has the freedom to set the scheme for any/all PAM-aware applications on your Linux system. That is, he can authenticate from anything as naive as simple trust (pam_permit) to something as paranoid as a combination of a retinal scan, a voice print and a one-time password!

Linux-PAM deals with four separate types of (management) task. These are: authentication management; account management; session management; and password management. The association of the preferred management
scheme with the behavior of an application is made with entries in the relevant Linux-PAM configuration file. The management functions are performed by modules specified in the configuration file.

Here is a figure that describes the overall organization of Linux-PAM.

By way of explanation, the left of the figure represents the application; application X. Such an application interfaces with the Linux-PAM library and knows none of the specifics of its configured authentication method. The Linux-PAM library (in the center) consults the contents of the PAM configuration file and loads the modules that are appropriate for application-X. These modules fall into one of four management groups (lower-center) and are stacked in the order they appear in the configuration file. These modules, when called by Linux-PAM, perform the
various authentication tasks for the application. Textual information, required from/or offered to the user, can be exchanged through the use of the application-supplied conversation function.

The Linux-PAM Configuration File

Linux-PAM is designed to provide the system administrator with a great deal of flexibility in configuring the privilege granting applications of their system. The local configuration of those aspects of system security controlled by Linux-PAM is contained in one of two places: either the single system file, /etc/pam.conf; or the /etc/pam.d/ directory. In this section we discuss the correct syntax of and generic options respected by entries to these files.

Configuration file syntax

The reader should note that the Linux-PAM specific tokens in this file are case insensitive. The module paths, however, are case sensitive since they indicate a file’s name and reflect the case dependence of typical Linux file-systems. The case-sensitivity of the arguments to any given module is defined for each module in turn.

In addition to the lines described below, there are two special characters provided for the convenience of the system administrator: comments are preceded by a ‘#’ and extend to the next end-of-line; also, module specification lines may be extended with a `\` escaped new-line.

A general configuration line of the /etc/pam.conf file has the following form:

Service-name module-type control-flag module-path arguments

Below, we explain the meaning of each of these tokens. The second (and more recently adopted) way of configuring Linux-PAM is via the contents of the /etc/pam.d/ directory. Once we have explained the meaning of the above tokens, we will describe this method.
Service-name

The name of the service associated with this entry. Frequently the service name is the conventional name of the given application. For example, ‘ftpd’, ‘rlogind’, ‘su’, etc.

There is a special service-name, reserved for defining a default authentication mechanism. It has the name ‘OTHER’ and may be specified in either lower or upper case characters. Note, when there is a module specified for a named service, the ‘OTHER’ entries are ignored.

Module-type

One of (currently) the four types of module. The four types are as follows:

Auth - this module type provides two aspects of authenticating the user. Firstly, it establishes that the user is who they claim to be, by instructing the application to prompt the user for a password or other means of identification. Secondly, the module can grant group membership, independently of the /etc/groups, or other privileges through its credential granting properties.

Account - this module performs non-authentication based account management. It is typically used to restrict/permit access to a service based on the time of day, currently available system resources (maximum number of users) or perhaps the location of the applicant user—`root` login only on the console.

Session - primarily, this module is associated with doing things that need to be done for the user before/after they can be given service. Such things include the logging of information concerning the opening/closing of some data exchange with a user, mounting directories, etc.

Password - this last module type is required for updating the authentication token associated with the user. Typically, there is one module for each ‘challenge/response’ based authentication (auth) module-type.
Control-flag

The control-flag is used to indicate how the PAM library will react to the success or failure of the module it is associated with. Since modules can be stacked (modules of the same type execute in series, one after another), the control-flags determine the relative importance of each module. The application is not made aware of the individual success or failure of modules listed in the ‘/etc/pam.conf’ file. Instead, it receives a summary success or fail responses from the Linux-PAM library. The order of execution of these modules is that of the entries in the /etc/pam.conf file: earlier entries are executed before later ones. The control-flag can be defined with one of two syntaxes. The simpler (and historical) syntax for the control-flag is a single keyword defined to indicate the severity of concern associated with the success or failure of a specific module. There are four such keywords: required, requisite, sufficient and optional.

The Linux-PAM library interprets these keywords in the following manner:

**Required** - this indicates that the success of the module is required for the module-type facility to succeed. Failure of this module will not be apparent to the user until all of the remaining modules (of the same module-type) have been executed.

**Requisite** - like required, however, in the case that such a module returns a failure, control is directly returned to the application. The return value is that associated with the first required or requisite module to fail. Note, this flag can be used to protect against the possibility of a user getting the opportunity to enter a password over an unsafe medium. It is conceivable that such behavior might inform an attacker of valid accounts on a system. This possibility should be weighed against the not insignificant concerns of exposing a sensitive password in a hostile environment.

**Sufficient** - the success of this module is deemed ‘sufficient’ to satisfy the Linux-PAM library that this module-type has succeeded in its purpose. In the event that no previous required module has failed, no more ‘stacked’ modules of this type are invoked. (Note, in this case subsequent required modules are not invoked.). A failure of this module is not deemed as fatal to satisfying the application that this module-type has succeeded.
**Optional** - as its name suggests, this control-flag marks the module as not being critical to the success or failure of the user’s application for service. In general, Linux-PAM ignores such a module when determining if the module stack will succeed or fail. However, in the absence of any definite successes or failures of previous or subsequent stacked modules this module will determine the nature of the response to the application. One example of this latter case, is when the other modules return something like PAM_IGNORE.

**Newest Syntax**
The more elaborate (newer) syntax is much more specific and gives the administrator a great deal of control over how the user is authenticated. This form of the control flag is delimited with square brackets and consists of a series of value=action tokens:

```plaintext
[value1=action1 value2=action2 ...]
```

Here, value1 is one of the following return values: success; open_err; symbol_err; service_err; system_err; buf_err; perm_denied; auth_err; cred_insufficient; authinfo_unavail; user_unknown; maxtries; new_authtok_reqd; acct_expired; session_err; cred_unavail; cred_expired; cred_err; no_module_data; conv_err; authtok_err; authtok_recover_err; authtok_lock_busy; authtok_disable_aging; try_again; ignore; abort; authtok_expired; module_unknown; bad_item; and default. The last of these (default) can be used to set the action for those return values that are not explicitly defined.

The action can be a positive integer or one of the following tokens: ignore; ok; done; bad; die; and reset.

- **A positive integer** - when specified as the action, can be used to indicate that the next J modules of the current type will be skipped. In this way, the administrator can develop a moderately sophisticated stack of modules with a number of different paths of execution. Which path is taken can be determined by the reactions of individual modules.

- **Ignore** - when used with a stack of modules, the module’s return status will not contribute to the return code the application obtains.
**Bad** - this action indicates that the return code should be thought of as indicative of the module failing. If this module is the first in the stack to fail, its status value will be used for that of the whole stack.

**Die** - equivalent to bad with the side effect of terminating the module stack and PAM immediately returning to the application.

**Ok** - this tells PAM that the administrator thinks this return code should contribute directly to the return code of the full stack of modules. In other words, if the former state of the stack would lead to a return of PAM_SUCCESS, the module’s return code will override this value. Note, if the former state of the stack holds some value that is an indicative of a module failure, this ‘ok’ value will not be used to override that value.

**Done** - equivalent to ok with the side effect of terminating the module stack and PAM immediately returning to the application.

**Reset** - clear all memory of the state of the module stack and start again with the next stacked module.

**Module-path**

The path-name of the dynamically loadable object file, the pluggable module itself. If the first character of the module path is ‘/’, it is assumed to be a complete path. If this is not the case, the given module path is appended to the default module path: /lib/security.
Currently the Cyclades-TS has the following modules available:

**pam_access** - Provides logdaemon style login access control.

**pam_deny** - Deny access to all users.

**pam_env** - This module allows the (un)setting of environment variables. Supported is the use of previously set environment variables as well as PAM_ITEMs such as PAM_RHOST.

**pam_filter** - This module was written to offer a plug-in alternative to programs like ttysnoop (XXX - need a reference). Since writing a filter that performs this function has not occurred, it is currently only a toy. The single filter provided with the module simply transposes upper and lower case letters in the input and output streams. (This can be very annoying and is not kind to termcap based editors).

**pam_group** - This module provides group-settings based on the user’s name and the terminal they are requesting a given service from. It takes note of the time of day.

**pam_issue** - This module presents the issue file (/etc/issue by default) when prompting for a username.

**pam_lastlog** - This session module maintains the /var/log/lastlog file. Adding an open entry when called via the pam_open_session() function and completing it when pam_close_session() is called. This module can also display a line of information about the last login of the user. If an application already performs these tasks, it is not necessary to use this module.

**pam_limits** - This module, through the Linux-PAM open-session hook, sets limits on the system resources that can be obtained in a user-session. Its actions are dictated more explicitly through the configuration file discussed below.

**pam_listfile** - The list-file module provides a way to deny or allow services based on an arbitrary file.
pam_motd - This module outputs the motd file (/etc/motd by default) upon successful login.

pam_nologin - Provides standard Unix nologin authentication.

pam_permit - This module is very dangerous. It should be used with extreme caution. Its action is always to permit access. It does nothing else.

pam_radius – Provides Radius server authentication and accounting.

pam_rootok - This module is for use in situations where the superuser wishes to gain access to a service without having to enter a password.

pam_securetty - Provides standard Unix securetty checking.

pam_time - Running a well regulated system occasionally involves restricting access to certain services in a selective manner. This module offers some time control for access to services offered by a system. Its actions are determined with a configuration file. This module can be configured to deny access to (individual) users based on their name, the time of day, the day of week, the service they are applying for and their terminal from which they are making their request.

pam_tacplus – Provides Tacacs+ Server authentication, authorization (account management), and accounting (session management)

pam_unix - This is the standard Unix authentication module. It uses standard calls from the system’s libraries to retrieve and set account information as well as authentication. Usually this is obtained from the etc/passwd and the /etc/shadow file as well if shadow is enabled.

pam_warn - This module is principally for logging information about a proposed authentication or application to update a password.
pam_wheel - Only permit root authentication to members of wheel group.

Arguments

The arguments are a list of tokens that are passed to the module when it is invoked. They are much like arguments to a typical Linux shell command. Generally, valid arguments are optional and are specific to any given module. Invalid arguments are ignored by a module, however, when encountering an invalid argument, the module is required to write an error to syslog(3).

The following are optional arguments which are likely to be understood by any module. Arguments (including these) are in general optional.

- **debug** - Use the syslog(3) call to log debugging information to the system log files.
- **no_warn** - Instruct module to not give warning messages to the application.
- **use_first_pass** - The module should not prompt the user for a password. Instead, it should obtain the previously typed password (from the preceding auth module), and use that. If that doesn’t work, then the user will not be authenticated. (This option is intended for auth and password modules only).
- **try_first_pass** - The module should attempt authentication with the previously typed password (from the preceding auth module). If that doesn’t work, then the user is prompted for a password. (This option is intended for auth modules only).
- **use_mapped_pass** - This argument is not currently supported by any of the modules in the Linux-PAM distribution because of possible consequences associated with U.S. encryption exporting restrictions.
- **expose_account** - In general the leakage of some information about user accounts is not a secure policy for modules to adopt. Sometimes information such as user names or home directories, or preferred shell, can
be used to attack a user’s account. In some circumstances, however, this sort of information is not deemed a threat: displaying a user’s full name when asking them for a password in a secured environment could also be called being ‘friendly’. The expose_account argument is a standard module argument to encourage a module to be less discrete about account information as it is deemed appropriate by the local administrator.

Any line in (one of) the configuration file(s), that is not formatted correctly, will generally tend (erring on the side of caution) to make the authentication process fail. A corresponding error is written to the system log files with a call to syslog(3).

**Directory based configuration**

More flexible than the single configuration file, it is possible to configure libpam via the contents of the /etc/pam.d/ directory. In this case the directory is filled with files each of which has a filename equal to a service-name (in lower-case): it is the personal configuration file for the named service.

The Cyclades-TS Linux-PAM was compiled to uses both /etc/pam.d/ and /etc/pam.conf in sequence. In this mode, entries in /etc/pam.d/ override those of /etc/pam.conf.

The syntax of each file in /etc/pam.d/ is similar to that of the /etc/pam.conf file and is made up of lines of the following form:

```
module-type control-flag module-path arguments
```

The only difference being that the service-name is not present. The service-name is of course the name of the given configuration file. For example, /etc/pam.d/login contains the configuration for the login service.
Example configuration file entries

This section gives some examples of entries that can be present in the Linux-PAM configuration file. As a first attempt at configuring your system you could do worse than to implement these.

Default policy

If a system is to be considered secure, it had better have a reasonably secure ‘OTHER’ entry. The following is a paranoid setting (which is not a bad place to start!):

```
# default; deny access
#
OTHER auth     required      pam_deny.so
OTHER account  required      pam_deny.so
OTHER password required      pam_deny.so
OTHER session  required      pam_deny.so
```

Whilst fundamentally a secure default, this is not very sympathetic to a misconfigured system. For example, such a system is vulnerable to locking everyone out should the rest of the file become badly written.

The module pam_deny not very sophisticated. For example, it logs no information when it is invoked so unless the users of a system contact the administrator when failing to execute a service application, the administrator may go for a long while in ignorance of the fact that his system is misconfigured.

The addition of the following line before those in the above example would provide a suitable warning to the administrator.
Having two “OTHER auth” lines is an example of stacking.

On a system that uses the /etc/pam.d/ configuration, the corresponding default setup would be achieved with the following file:

```
# default configuration: /etc/pam.d/other
#
auth  required  pam_warn.so
auth  required  pam_deny.so
account required  pam_deny.so
password required  pam_warn.so
password required  pam_deny.so
session required  pam_deny.so
```

On a less sensitive computer, one on which the system administrator wishes to remain ignorant of much of the power of Linux-PAM, the following selection of lines (in /etc/pam.conf) is likely to mimic the historically familiar Linux setup.

```
# default; standard UNIX access
#
OTHER auth required  pam_unix_auth.so
OTHER account required  pam_unix_acct.so
```
In general this will provide a starting place for most applications.

**Cyclades-TS Default pam.conf file**

In addition to the normal applications login, su, sshd, passwd, and pppd Cyclades also has made portslave a PAM-aware application. The portslave requires four services configured in the pam.conf. They are local, remote, radius, and tacplus. The portslave PAM interface takes any parameter needed to perform the authentication in the serial ports from the file pslave.conf. The pslave.conf parameter all.authtype determines which service(s) should be used.

```
# /etc/pam.conf
#
#
# Last modified by Andrew G. Morgan <morgan@kernel.org>
#
# $Id: pam.conf,v 1.2 2001/04/08 06:02:33 agmorgan Exp $
#
# serv. module   ctrl   module [path] ...[args..]
#
# name type flag
#
# ____________________________________________________________#
```
# The PAM configuration file for the 'tacplus' service
#
tacplus auth    requisite    pam_securetty.so
    tacplus auth    required    pam_tacplus.so encrypt
    tacplus account required    pam_tacplus.so encrypt service=ppp protocol=lcp
    tacplus session required    pam_tacplus.so encrypt service=ppp protocol=lcp

# The PAM configuration file for the 'radius' service
#
    radius auth    requisite    pam_securetty.so
    radius auth    required    pam_radius_auth.so
    radius account required    pam_radius_auth.so
    radius session required    pam_radius_auth.so

# The PAM configuration file for the 'local' service
#
    local auth    requisite    pam_securetty.so
    local auth    required    pam_unix.so
    local account required    pam_unix.so
    local password required    pam_unix.so md5 use_authtok
    local session required    pam_unix.so

# The PAM configuration file for the 'remote' service
#
    remote auth    required    pam_permit.so
    remote account required    pam_permit.so
remote password    required  pam_permit.so
remote session     required  pam_permit.so

#  
# The PAM configuration file for the ‘login’ service  
#  
login auth         requisite  pam_securetty.so
login auth         required  pam_unix.so
login auth         optional   pam_group.so
login account      requisite  pam_time.so
login account      required  pam_unix.so
login password     required  pam_unix.so md5 use_authtok
login session      required  pam_unix.so
login session      required  pam_limits.so

#  
# The PAM configuration file for the ‘xsh’ service  
#  
sshd auth          required  pam_unix.so
sshd auth          optional   pam_group.so
sshd account       requisite  pam_time.so
sshd account       required  pam_unix.so
sshd password      required  pam_unix.so md5 use_authtok
sshd session       required  pam_unix.so
sshd session       required  pam_limits.so

#  
# The PAM configuration file for the ‘passwd’ service  
#  
passwd password    required  pam_unix.so md5  
#  
Appendix J  Linux-PAM
# The PAM configuration file for the 'samba' service
#
samba auth      required   pam_unix.so
samba account   required   pam_unix.so
#
# The PAM configuration file for the 'su' service
#
su auth         required   pam_wheel.so
su auth         sufficient pam_rootok.so
su auth         required   pam_unix.so
su account      required   pam_unix.so
su session      required   pam_unix.so
#
# Information for the PPPD process with the 'login' option.
#
ppp auth         required   pam_nologin.so
ppp auth         required   pam_unix.so
ppp account      required   pam_unix.so
ppp session      required   pam_unix.so
#
# The PAM configuration file for the 'other' service
#
other auth       required   pam_warn.so
other auth       required   pam_deny.so
other account    required   pam_deny.so
other password   required   pam_warn.so
other password required pam_den.so
other session required pam_den.so

Reference

The Linux-PAM System Administrators’ Guide

Copyright (c) Andrew G. Morgan 1996-9. All rights reserved.
Email: morgan@linux.kernel.org
APPENDIX K TIMEZONE

The content of the file /etc/TIMEZONE can be one of two formats. The first format is used when there is no daylight saving time in the local time zone:

    std offset

The std string specifies the name of the time zone and must be three or more alphabetic characters. The offset string immediately follows std and specifies the time value to be added to the local time to get Coordinated Universal Time (UTC). The offset is positive if the local time zone is west of the Prime Meridian and negative if it is east. The hour must be between 0 and 24, and the minutes and seconds 0 and 59.

The second format is used when there is daylight saving time:

    std offset dst [offset],start[/time],end[/time]

There are no spaces in the specification. The initial std and offset specify the standard time zone, as described above. The dst string and offset specify the name and offset for the corresponding daylight savings time zone. If the offset is omitted, it defaults to one hour ahead of standard time.

The start field specifies when daylight savings time goes into effect and the end field specifies when the change is made back to standard time. These fields may have the following formats:

- Jn This specifies the Julian day with n between 1 and 365. February 29 is never counted even in leap years.
- n This specifies the Julian day with n between 1 and 365. February 29 is counted in leap years.
- Mm.w.d This specifies day d (0 <= d <= 6) of week w (1 <= w <= 5) of month
m (1 <= m <= 12). Week 1 is the first week in which day d occurs and week 5 is the last week in which day d occurs. Day 0 is a Sunday.

The time fields specify when, in the local time currently in effect, the change to the other time occurs. If omitted, the default is 02:00:00.

In the example below:

GST+7DST+6M4.1.0/14:30.M10.5.6/10

The daylight saving time starts on the first Sunday of April at 2:30 pm and it ends on the last Saturday of October at 10:00 am.
Cyclades Corporation
41829 Albrae Street
Fremont, CA 94538 - USA
Phone: (510) 770-9727
Fax: (510) 770-0355
www.cyclades.com

Cyclades Australia
Phone: +61 7 3279 4320
Fax: +61 7 3279 4393
www.au.cyclades.com

Cyclades Philippines
Phone: (632) 813-0353
Fax: (632) 655-2610
www.ph.cyclades.com

Cyclades UK
Phone: +44 1724 277179
Fax: +44 1724 279981
www.uk.cyclades.com

Cyclades South America
Phone: 55-11-5033-3333
Fax: 55-11-5033-3388
www.cyclades.com.br

Cyclades Italy
Phone: 39 329 0990451

Cyclades Germany
Phone: +49 (0)81 22 90 99-90
Fax: +49 (0)81 22 90 999-33
www.cyclades.de