Connectware™

PortServer CM™

User Manual

PortServer CM Installation Manual
Version 1.0 – January 2002
Copyright (C) Digi International Inc., 2002

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 Manual.

This manual is published by Digi International Inc., 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.

All brand and product names mentioned in this publication are trademarks or registered trademarks of their respective holders.

FCC Warning Statement:
The PortServer CM 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 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.

Canadian DOC Notice:
The **PortServer** CM 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 **PortServer** CM n'émete pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe A prescrites dans le règlement sur le brouillage radioélectrique edicté par le Ministère des Communications du Canada.

Table of Contents

CHAPTER 1 HOW TO USE THIS MANUAL 5
CHAPTER 2 SAFETY INSTRUCTIONS 6
Using Your PortServer CM 6
Working Inside the PortServer CM 7
Replacing the Battery 7
CHAPTER 3 WHAT IS IN THE BOX 8
CHAPTER 4 SUMMARY OF THE CONFIGURATION PROCESS 9
CHAPTER 5 CONFIGURATION 11
STEP ONE 11
STEP TWO 11
STEP THREE 13
STEP FOUR 21
Clustering 22
CHAPTER 6 UPGRADES AND TROUBLESHOOTING 27
Upgrades 27
Troubleshooting 27
Hardware Test 29
To Change the root Password 31
APPENDIX A INFORMATION FOR USERS NOT FAMILIAR WITH LINUX 33
Users and Passwords 33
Linux File Structure 33
Basic File Manipulation Commands 34
The vi Editor 35
The Routing Table 36
ssh - The Secure Shell Session 37
The Process Table 39
NTP Client Functionality 39

The Crond Utility 39
The DHCP (Dynamic Host Configuration Protocol) Client 40
Packet Filtering using ipchains 42
An example of the use of ipchains for a console access server *44*
Using digi_menu to simplify port connections 44
APPENDIX B HARDWARE SPECIFICATIONS 46
APPENDIX C SAMPLE PSLAVE.CONF FILES 54
The Complete pslave.conf File Provided with the PortServer CM 54
APPENDIX D CUSTOMIZATION 66
APPENDIX E THE WEB CONFIGURATION MANAGER 67
Troubleshooting the Web Configuration Manager *73*

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 PortServer CM is a Linux-based secure console access 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 PortServer CM. The files can be edited in the PortServer CM 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 manual before jumping in.

This manual should be read in the order written, with exceptions given in the text.

Chapter 2 - Safety Instructions **-** contains important saftey instructions for operating the PortServer CM.
Chapter 3 - What is in the Box **-** explains how the PortServer CM should be connected and what each cable is used for.
Chapter 4 - Summary of the Configuration Process - provides a brief roadmap of the PortServer CM configuration.
Chapter 5 - Configuration **-** describes the basic configuration process to get the PortServer CM up and running for the most common uses.
Chapter 6 - Troubleshooting - provides solutions and test procedures for typical problems.
Appendix A - Linux Information - Information for those who are new to Linux/UNIX.
Appendix B - Hardware Specifications - Pinout diagrams for cables.
Appendix C - Pslave.conf Sample Files - Example files for 3 profiles and the master file.
Appendix D - Customization - Instructions for those who wish to create their own applications.
Appendix E - The Web Configuration Manager - explains how to configure the CM with a browser.

CHAPTER 2 SAFETY INSTRUCTIONS

Use the following safety guidelines to protect yourself and your PortServer CM™.

Using Your PortServer CM

CAUTION: Do not operate your PortServer CM with the cover removed.

- In order to avoid shorting out your PortServer CM 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 PortServer CM into properly grounded power source. The cable is equipped with 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 PortServer CM from transients in electrical power, use a surge suppressor, line conditioner, or uninterruptible power supply.

- Be sure that nothing rests on PortServer CM' cables and that the cables are not located where they can be stepped on or tripped over.

- Do not spill food or liquids on your PortServer CM. If it gets wet, contact Digi.

- Do not push any objects into the openings of your PortServer CM. Doing so can cause fire or electric shock by shorting out interior components.

- Keep your PortServer CM away from heat sources. Also, do not block cooling vents.

Working Inside the PortServer CM

NOTICE: Do not attempt to service the PortServer CM yourself, except following the instructions from Digi Technical Support personnel. In such a case, perform first the following actions:

- Turn off the PortServer CM.
- Ground yourself by touching an unpainted metal surface at the back of the equipment before touching anything inside your equipment.

Replacing the Battery

A coin-cell battery maintains date and time information. If you have to repeatedly reset time and date information after turning on your PortServer CM, 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 following figure shows the main unit, accessories included in the 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.

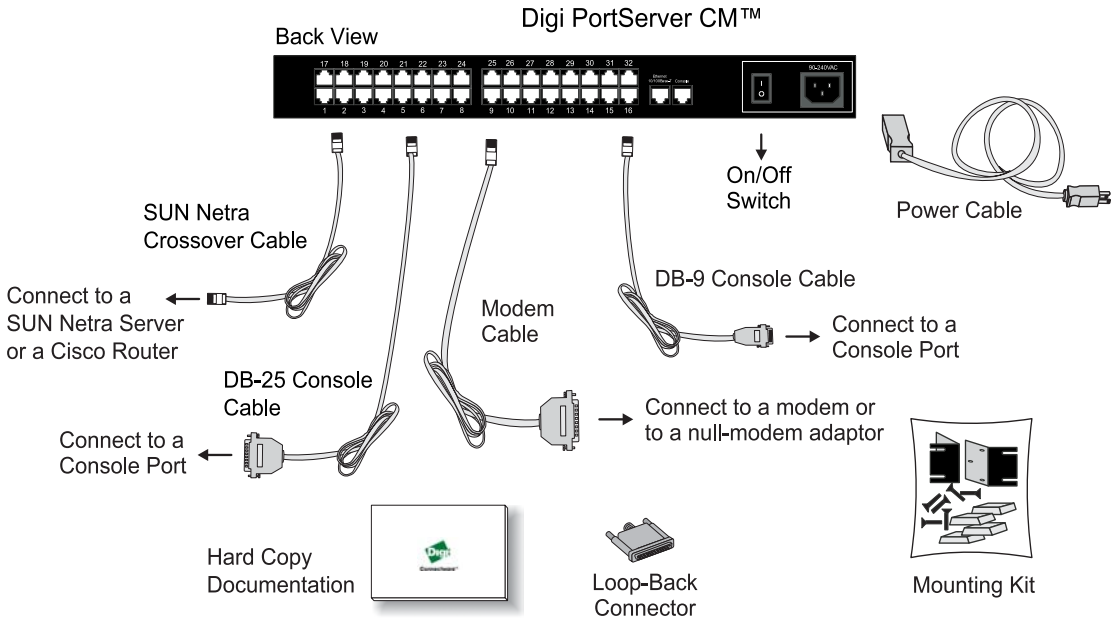


FIGURE 3.1 PORTSERVER CM AND CABLES

CHAPTER 4 SUMMARY OF THE CONFIGURATION PROCESS

The PortServer CM'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 PortServer CM. Additional information about the files and tools needed for configuration is provided in appendix A.

Initial configuration steps are:

A. Connecting the PortServer CM 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 PortServer CM via the console port and login.

C. Configure the network interface of the PortServer CM. Consult Chapter 5, Configuration, for details.

D. Configure the boot parameters using the command bootconf.

E. Edit the pslave.conf file. This is the main configuration file that concentrates most product parameters and defines the functionality of the PortServer CM. 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.

CHAPTER 5 CONFIGURATION

This chapter guides you step by step through the configuration of the PortServer CM.

STEP ONE

Connect a PC or terminal to the PortServer CM using the console cable. If using a PC, HyperTerminal can be used in the Windows operating system or Kermit 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: Hardware flow control or none
- Ansi emulation (Note: if your terminal does not have ansi emulation, select vt100; then, on the CM, log in as root and switch to vt100 by typing “TERM=vt100;export TERM”)

When the PortServer CM boots properly, you will see a series of messages displayed as the unit loads each operating system component followed by a login banner.

Log in as *root* (there is no password). A password should be created as soon as possible. The PortServer CM 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 the Linux appendix at the end of this manual.

STEP TWO

In this step, four Linux files must be modified to identify the PortServer CM and its neighbors. Analternative to editing each file is to use the `netconfig` command. Then, the boot parameters are configured. The OS provides a scaled-down version of the vi editor. A description of it's features is available in the Linux appendix. The first file is /etc/hostname. The only entry should be the hostname of the PortServer CM. An example is shown in Figure 5.1.

```
DigiPSCM
```

FIGURE 5.1 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    DigiPSCM
 200.200.200.2    RADIUSServer
 127.0.0.1        localhost
```

FIGURE 5.2 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 5.3 CONTENTS OF THE /ETC/RESOLV.CONF FILE

The fourth file defines static routes and is called /etc/network/st_routes. The IP address of your network gateway router should be configured in this file. Other static routes are also configured in this file.

```
route add default gw   200.200.200.5
```

FIGURE 5.4 CONTENTS OF THE /ETC/NETWORK/ST_ROUTES FILE

STEP THREE

A CM application is shown in Figure 5.5.

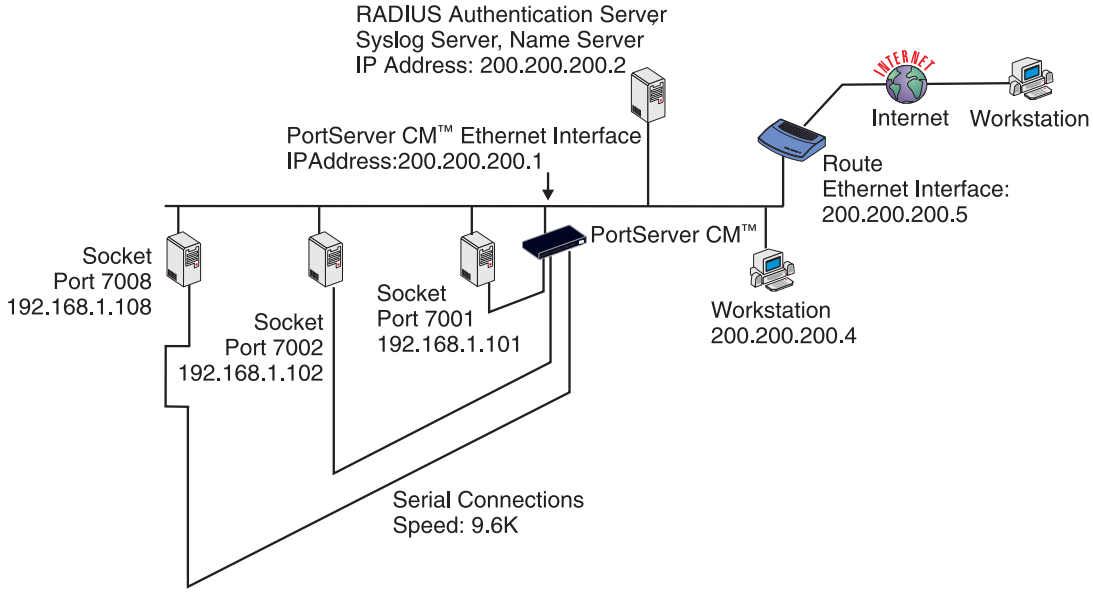


FIGURE 5.5 CM APPLICATION

This application allows a user to access a device connected to the PortServer CM through its serial console port from a workstation on the LAN or WAN. A console session 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 infomation about ssh.

The file, /etc/portslave/pslave.conf, is specific to the PortServer CM and a sample file with comments is supplied in the Linux file system. It is called /etc/portslave/pslave.conf. A listing of the pslave.conf file with all possible parameters 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 overriden by a s#.* parameter appearing later in the pslave.conf file (or vice-versa). A brief description of each parameter used for the CM profile is given in Figures 5.6-5.7.

Parameter	Description	Value for This Example
conf.eth_ip	The IP address of the Ethernet interface. This parameter, along with the next two, is used by the digi_ras program to OVERWRITE the file /etc/network/ifcfg_eth0 as soon as the command "signal_ras hup" is executed. The file /etc/network/ifcfg_eth0 should not be edited by the user unless the digi_ras application is not going to be used. You may use an alternative command, “netconfig”, to configure network parameters.	200.200.200.1
conf.eth_mask	The mask for the Ethernet network. You may use an alternative command, “netconfig”, to configure network parameters.	255.255.255.0
conf.eth_mtu	The Maximum Transmission Unit size, which determines whether or not packets should be broken up.	1500
conf.nfs_data_ buffering	Remote Network File System where data buffering will be written instead of the default directory '/var/run'. 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 in the directory and server indicated by this variable. 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.	commented
conf.lockdir	The lock directory , which is /var/lock for the PortServer CM. It should not be changed unless the user decides to customize the operating system.	/var/lock

FIGURE 5.6 CM PSLAVE.CONF GLOBAL PARAMETERS

Parameter	Description	Value for This Example
conf.syslog	The IP address of a remote syslog daemon can be provided here, if desired.	200.200.200.2
conf.facility	This value (0-7) is sent to the syslog server (the CM is a syslog client) to indicate in which file the syslog messages sent by portslave should be stored. The file /etc/syslog.conf on the syslog server contains a mapping between facility numbers and server log files.	7
conf.DB_facility	This value (0-7) is sent to the syslog server (the CM is a syslog client) to indicate in which file the syslog messages sent by the data buffering feature should be stored. See description for conf.facility.	0

FIGURE 5.6 CM PSLAVE.CONF GLOBAL PARAMETERS (CONT.)

Parameter	Description	Value in Exp.
all.syslog_level	This variable determines which syslog messages will be sent to the syslog server configured in the conf.syslog parameter. A value of 0 suppresses all but emergency messages while values between 1 and 7 send progressively more types of messages for each increment. **This value (as for all "all." parameters) can later be overridden for individual ports using the s<*port number*>.syslog_level parameter.**	4
all.console_level	This variable determines which syslog messages will be sent to the PortServer CM console connected through the console interface. See the previous parameter for a description of possible values	4
all.speed	The speed for all ports.	9600
all.datasize	The data size for all ports.	8
all.stopbits	The number of stop bits for all ports	1
all.parity	The parity for all ports.	none

FIGURE 5.7 CM 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), none, local/radius (authentication is performed locally first, switching to RADIUS if unsuccessful), radius/local (the opposite of the previous option) and RADIUSDownLocal (local authentication is tried only when the RADIUS server is down). Note that this parameter controls the authentication required by the PortServer CM. 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 authentication server and is only necessary if this option is chosen in the previous parameter. A second RADIUS authentication server can be configured with the parameter all.authhost2.	200.200.200.2
all.accthost1	This address indicates the location of the RADIUS accounting server, which can be used to track how long users are connected after being authorized by the authentication server. Its use is optional.	200.200.200.2
all.radtimeout	This is the timeout (in seconds) for a radius authentication query. The first server (authhost1) is tried "radretries" times, and then the second (if configured) is contacted "radretries" times. If the second also fails to respond, RADIUS authentication fails.	3
all.radretries	Defines the number of times each RADIUS server is tried before another is contacted. The default, if not configured, is 5.	5
all.secret	This is the shared secret necessary for communication between the PortServer CM and the RADIUS servers.	digi

FIGURE 5.7 CM PSLAVE.CONF PORT-SPECIFIC PARAMETERS(CONT.)

Parameter	Description	Value for This Example
all.ipno	This is the default IP address of the PortServer CM's serial ports. The "+" 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.	192.168.1.101+
all.issue	This text determines the format of the login banner that is issued when a connection is made to the PortServer CM. \n represents a new line and \r represents a carriage return.	\r\n\ PortServer CM 32\n\ \r\nWelcome to %h port %p \n\ \r\n
all.prompt	This text defines the format of the login prompt. Expansion characters, listed in Appendix C, can be used here.	%h login:
all.flow	This sets the flow control to hardware, software, or none.	hard
all.poll_interval	When not set to zero, this parameter sets the wait for a TCP connection keep-alive timer. If no traffic passes through the PortServer CM for this period of time, the PortServer CM will send a modem status message to the remote device to see if the connection is still up.	0
all.socket_port	This defines an alternative labeling system for the PortServer CM 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.	7001+
all.protocol	For the CM profile, the possible protocols are socket_server (when telnet is used) and socket_ssh (when ssh version one or two is used).	socket_server

FIGURE 5.7 CM PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONT.)

Parameter	Description	Value for This Example
all.data_buffering	A non-zero value activates data buffering. A file (/var/run/ttyS#.data) is created on the PortServer CM and all data received from the port is captured. The file contains a maximum size equal to this parameter, which is overwritten each time the maximum is reached. This file can be viewed using the normal UNIX tools (cat, vi, more, etc.).	0
all.syslog_buffering	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. The syslog level for data buffering is hard coded to level 5, so the parameter syslog_level should be greater than or equal to 5, and data_buffering non-zero for syslog generation.	0
all.dont_show_DB menu	When zero, shows a menu with data buffering options when a non-empty data buffering file is found.	1
all.users	Restricts access to ports by user name (only the users listed can access the port or all but the users listed can access the port (with !).) 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.	! joe, mark
all.sniff_mode	This parameter determines what the second connected user (see parameter admin_users below) can see of the session of the first connected user: *in* shows data written to the port, *out* shows data received from the port, and *i/o* shows both streams. The second session is called a sniff session and this feature is activated whenever the protocol parameter is set to socket_ssh or socket_server.	out

FIGURE 5.7 CM PSLAVE.CONF PORT-SPECIFIC PARAMETERS (CONT.)

Parameter	Description	Value for This Example
all.admin_users	This parameter determines which users can open a *sniff session*, which is where a second connected user can see everything that a first connected user is doing on a given port. The second user can also cancel the first user's session (and take over). Only two users can connect to the same port simultaneously. This parameter is obligatory when *authtype* is not *none*, to determine who can open a sniff session or cancel a previous session.	peter, john
s1.tty	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.	ttyS1
s1.authtype	Authtype must not be none for the *sniff session* feature to function with authentication. If none is chosen, any user can open a sniff session and/or cancel sessions of other users.	local
s2.tty	See the s1.tty entry in this table.	ttyS2
s8.tty	See the s1.tty entry in this table.	ttyS8

FIGURE 5.7 CM 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 PortServer CM 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.
5. The PortServer CM has been set for communication at 9600 bps, 8N1. The device 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 the serial console port.

5. From a server on the LAN (not from the console), try to telnet to the device connected to the first port of the PortServer CM using the following command:

```
telnet 200.200.200.1 7001
```

For both telnet and ssh sessions, the devices can be reached by either:
1. Ethernet IP of the PortServer CM and assigned socket port
or
2. Individual IP assigned to each port.

If everything is configured correctly, a telnet session should open on the device 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.

STEP FOUR

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.

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. The previous contents of the file /proc/flash/script will be lost.

Restart the digi_ras process. This can be done by executing the command:

```
signal_ras hup
```

Now the configuration is complete.

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 PortServer CM is booted.

Clustering

Clustering allows the stringing of CMs so that one master PortServer CM can be used to access all PortServer CMs on a LAN. The master PortServer CM can manage up to 512 serial ports, so

- 1 Master PortServer CM + 15 slave PortServer CMs

can be clustered.

An example with one master PortServer CM and two slave PortServer CMs is shown in Figure 5.8.

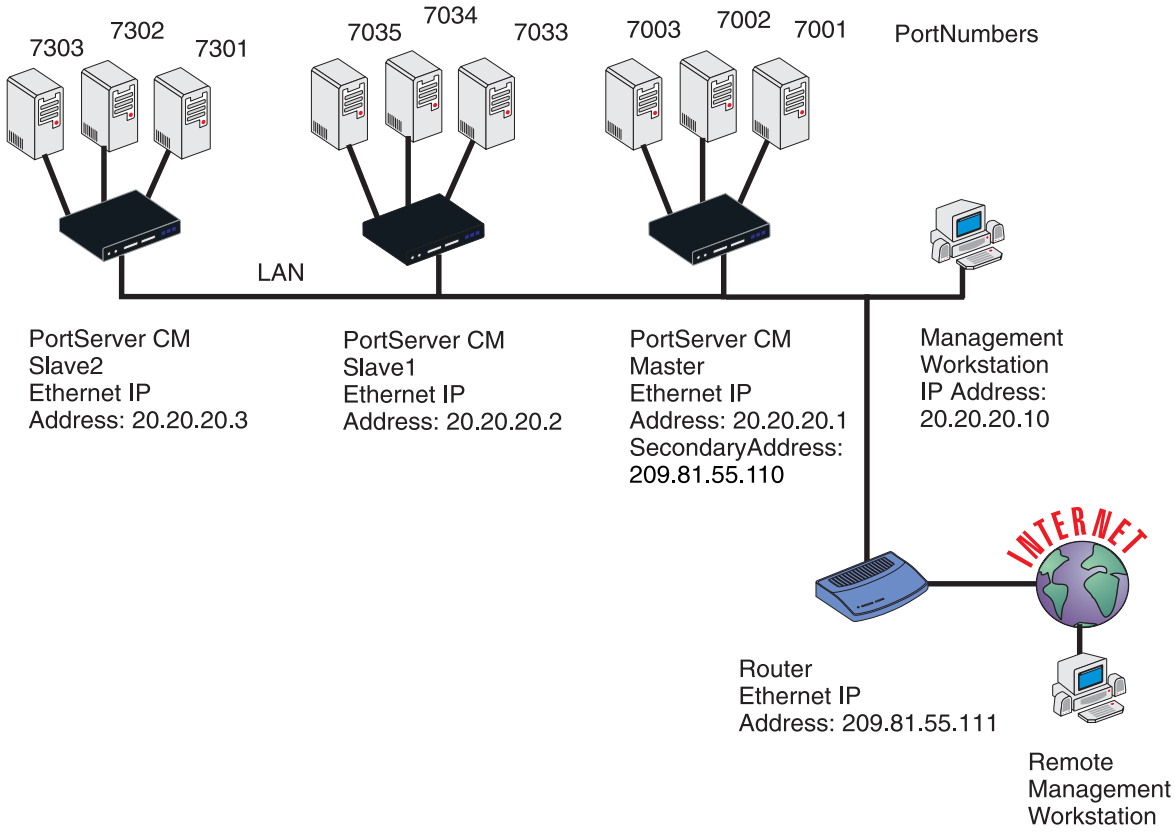


FIGURE 5.8 EXAMPLE USING THE CLUSTERING FEATURE.

The Master PortServer CM must contain references to the Slave ports. The configuration described earlier should be followed with the following exceptions for the Master and Slaves:

Master Configuration:

Parameter	Description	Value for This Example
conf.eth_ip	Ethernet Interface IP address.	20.20.20.1
conf.eth_ip_alias	Secondary IP address for the Ethernet Interface (needed for clustering feature).	209.81.55.110
conf.eth_mask_alias	Mask for secondary IP address above.	255.255.255.0
all.socket_port	This value applies to both the local ports and ports on slave PortServer CMs.	7001+
all.protocol	Depends on the application.	Socket_ssh or socket_server
all.authtype	Depends on the application.	RADIUS or local or none
s33.tty	This parameter must be created in the master CM file for every slave port. Its format is IP_of_Slave[:slave_socket_port] for non-master ports. 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.	20.20.20.2:7033
s33.serverfarm	An alias for this port.	Server_on_slave1_serial_s1
s33.ipno	This parameter must be created in the master CM file for every slave port, unless configured using all.ipno.	0.0.0.0
s34.tty	See s33.tty.	20.20.20.2:7034
s34.serverfarm	An alias for this port.	Server_on_slave1_serial_s2
s34.ipno	See s33.ipno.	0.0.0.0

FIGURE 5.9 MASTER PORTSERVER CM CONFIGURATION

Parameter	Description	Value for This Example
s35.tty	See s33.tty.	20.20.20.2:7035
s35.serverfarm	An alias for this port.	Server_on_slave1_serial_s3
s35.ipno	See s33.ipno.	0.0.0.0
etc. for s36-s64		
S65.tty	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.	20.20.20.3:7301
S65.serverfarm	An alias for this port.	Server_on_slave2_serial_s1
S65.ipno	See s33.ipno.	0.0.0.0
S66.tty	See s65.tty.	20.20.20.3:7302
S66.serverfarm	An alias for this port.	Server_on_slave2_serial_s2
S66.ipno	See s33.ipno.	0.0.0.0
S67.tty	See s65.tty.	20.20.20.3:7303
S67.serverfarm	An alias for this port.	Server_on_slave2_serial_s3
S67.ipno	See s33.ipno.	0.0.0.0
etc. for s68-s96		

FIGURE 5.9 MASTER PORTSERVER CM CONFIGURATION (CONT.)

The Slave PortServer CMs do not need to know they are being accessed through the Master PortServer CM. Their port numbers, however, must agree with those assigned by the Master.

Parameter	Value for This Example
all.protocol	socket_server
all.authtype	none
conf.eth_ip	20.20.20.2
all.socket_port	7033+

FIGURE 5.10 PORTSERVER CM CONFIGURATION FOR SLAVE 1

Parameter	Value for This Example
all.protocol	Socket_server
all.authtype	None
conf.eth_ip	20.20.20.3
all.socket_port	7301+

FIGURE 5.11 PORTSERVER CM CONFIGURATION FOR SLAVE 2

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 PortServer CM
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.

The `ssh` command 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
```

CHAPTER 6 UPGRADES AND TROUBLESHOOTING

Upgrades

The files added by Digi 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 PortServer CM configuration information is stored

To upgrade the Linux kernel provided in the PortServer CM, ftp the new zImage file on top of the zImage file in the /proc/flash directory.

```
[root@portserver_cm /root]# cd /proc/flash
[root@portserver_cm flash]# ftp [ftp server name]
[root@portserver_cm flash]# cd [directory containing zimage file]
[root@portserver_cm flash]# bin (change to binary mode)
[root@portserver_cm flash]# get zimage
```

Then, reboot, and 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 PortServer CM 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 5, 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.

File	Program
/etc/securetty	telnet, login, su
/etc/issue	getty
/etc/getty_ttyS0	login (via console)
/etc/hostname	tcp
/etc/hosts	tcp
/etc/host.conf	tcp
/etc/nsswitch.conf	dns
/etc/resolv.conf	dns
/etc/config_files	saveconf
/etc/passwd	login, passwd, adduser...
/etc/group	login, passwd, adduser...
/etc/ssh/ssh_host_key.pub	sshd
/etc/ssh/sshd_config	sshd
/etc/ssh/ssh_config	ssh client
/etc/ssh/ssh_host_key	sshd (ssh1)
/etc/ssh/ssh_host_key.pub	sshd (ssh1)
/etc/ssh/ssh_host_dsa_key	sshd (ssh2)
/etc/ssh/ssh_host_dsa_key.pub	sshd (ssh2)
/etc/snmp/snmpd.conf	snmpd
/etc/portslave/pslave.conf	digi_ras, portslave, CM configuration information
/etc/network/ifcfg_eth0	ifconfig eth0, digi_ras, rc.sysconf
/etc/network/ifcfg*	ifconfig, digi_ras, rc.sysinit
/etc/network/ifcfg_lo	ifconfig lo, digi_ras, rc.sysinit
/var/run/radsession.id	radinit, radius authentication process
/home	adduser, passwd
/etc/network/st_routes	ifconfig, digi_ras, rc.sysconf

If any of the files listed in /etc/config_files is modified, the PortServer CM administrator must execute the command saveconf before rebooting the PortServer CM 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.

Hardware Test

A hardware test called digitest is included with the PortServer CM firmware. It is a menu-driven program, run by typing digitest at the command prompt, and the various options are described below. **Note that the PortServer CM should not be tested while in use as the test will deactivate all ports.**

Port Test

Either a cross cable or a loop-back connector is necessary for this test. The 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). When digitest senses the presence of the cable or connector, the test will be automatically run and the result shown on the screen.

Each line of data correponds 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.

			<- Packets ->			<- Errors ->		
From		To	Sent	Received	Passes	Data	CTS	DCD	DSR
2	<->	2	35	35	35	0	0	0	0
4	<->	5	35	35	35	0	0	0	0
5	<->	4	35	35	35	0	0	0	0

When this test is run with a cable or connector without the DSR signal (see the pinout diagram in Appendix B 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, digitest 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, 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.

State	DTR	DCD	DSR	RTS	CTS
ON	X			X	
	↓			↓	
OFF		X	X		X

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.

State	DTR	DCD	DSR	RTS	CTS
ON	X	X	X	X	
	↓	↓	↓		
OFF					X

This is because the test is receiving the DTR signal sent through the DCD and DSR channels. 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.

To Change the root Password

The PortServer CM 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 PortServer CM unstable,
- After a configuration change which leaves the PortServer CM 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 00FFF596
avail ram: 0030B270 00E18000
Linux/PPC load: root=/dev/ram
```

After printing “Linux/PPC load: root=/dev/ram”, the PortServer CM waits approximately 10 seconds for user input. This is where the user should type “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 CM, 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.

APPENDIX A INFORMATION FOR USERS NOT FAMILIAR WITH LINUX

Users and Passwords

A username and password are necessary to log in to the PortServer CM. The user “root” is predefined, without a password. 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 priveleges), use the commands:

```
adduser user_name
passwd user_password
```

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:

/home	Contains the work directories of system users.
/bin	Contains applications and utilities used during system initialization.
/dev	Contains files for devices and ports.
/etc	Contains configuration files specific to the operating system.
/lib	Contains shared libraries.
/proc	Contains process information
/mnt	Contains information about mounted disks.
/opt	Location where packages not supplied with the operating system are stored.
/tmp	Location where temporary files are stored.
/usr	Contains most of the operating system files.
/var	Contains operating system data files.

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.

pwd	Supplies the name of the current directory. While logged in, the user is always "in" a directory. The default initial directory is the user's home directory, /home/<username>
ls [options] *directory_name*	Lists the files and directories within *directory_name*. Some useful options are -l for more detailed output and -a which shows hidden system files.
cd *directory_name*	Changes the directory to the one specified
cat *file_name*	Prints the contents of *file_name* to the screen.

Shortcuts:

. (a dot)	represents the current directory
.. (two dots)	represents one directory above the current directory (i.e. one directory closer to the base directory).

The vi Editor

To edit a file using the vi editor, type

```
vi [file name]
```

The vi utility 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.

Mode	What is done there	How to Get There
command mode	navigation within the open file	Press the <ESC> key.
editing mode	text editing	See list of editing commands below.
line mode	file saving, opening, etc. exiting from vi	From the command mode, type ":" (the colon).

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:

`h`	moves the cursor to the left (left arrow)
`j`	moves the cursor to the next line (down arrow)
`k`	moves the cursor to the previous line (up arrow)
`l`	moves the cursor to the right (right arrow)

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)

`i`	insert text before the cursor position (everything to the right of the cursor is shifted right)
`o`	create a new line below the current line and insert text (all lines are shifted down)
`dd`	remove the entire current line
`u`	undo the last modification
`x`	delete the letter at the cursor position

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

`w`	save the file (w is for write)
`wq`	save and close the file (q is for quit)
`q!`	close the file without saving
`w` *`file`*	save the file with the name *file*
`e` *`file`*	opens the file named *file*

The Routing Table

The PortServer CM 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 4.5 is the PortServer CM's method for configuring static routes. Routes should be added to the file (which is a script run when the PortServer CM is initialized) or at the prompt (for temporary routes) using the following syntax:

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

`[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` *`nt_msk`*	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.
`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 PortServer CM 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:	root
PortServer CM IP address:	192.168.160.1
host name:	cm32
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 root:ttyS1@cm32
ssh -t root:7001@cm32
ssh -t root:192.168.160.1@cm32
ssh -t root:file_server@cm32
ssh -t -l root:192.168.160.1
```

Note that either -l or @ are used, but not both. For ssh2, the -2 flag is used:

```
ssh -t -2 root:7001@cm32
```

To log in to a port that does not require authentication, the username is not necessary:

```
ssh -t -2 :ttyS1@cm32
```

The Process Table

The process table shows which processes are running. Type ps -a to see a table similar to that below.

```
PID     Uid       Gid      State    Command
1       root      root     S        /sbin/inetd
31      root      root     S        /sbin/sshd
32      root      root     S        /sbin/digi_ras
36      root      root     S        /sbin/digi_wdt_led wdt led
154     root      root     R        /ps -a
```

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

```
signal_ras hup
```

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

NTP Client Functionality

In order for the PortServer CM 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 /bin/ex_ntpclient

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 priviledges 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 tst_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. 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 4 in chapter 4 for more details.

The DHCP (Dynamic Host Configuration Protocol) Client

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 configured manually. 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.

To activate the DHCP client on the Ethernet Interface, set the global parameter conf.dhcp_client to 1. All other parameters related to the Ethernet Interface (conf.eth_ip, etc.) should be commented. If the IP addresses of the PortServer CM or the default gateway are changed, the PortServer CM will adjust its routing table accordingly. You may use an alternative command, “netconfig”, to configure network parameters.

To inactivate the DHCP client, set the parameter conf.dhcp_client to 0. Do not forget to uncomment the Ethernet parameters mentioned in the previous paragraph.

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 - command which activates the DHCP client (used by the digi_ras program). Its factory set contents are:
/sbin/dhcpcd -c /sbin/handle_dhcp
The dhcpcd command has other options which can be used on the command line. They 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.

Packet Filtering using ipchains

The PortServer CM 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 PortServer CM) or the howto: http://netfilter.filewatcher.org/ipchains/HOWTO.html.

The syntax of the ipchains command is:

```
ipchains -command chain [-s source] [-d destination] [-p protocol] [-j
target] [-i interface]
```

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 PortServer CM and are not destined to the PortServer CM

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 PortServer CM) 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 4.5
If the administrator wishes to restrict access to the consoles connected to the PortServer CM 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
ipchains -A input -p tcp -s ! 200.200.200.4 -d 200.200.200.1 7001:7032 -j DENY
ipchains -A input -p tcp -s ! 200.200.200.4 -d 0.0.0.0/0 22 -j DENY
```

Using digi_menu to simplify port connections

The digi_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 PortServer CM. The server is selected by its corresponding number.

Only ports configured for console access (protocols socket_server or socket_ssh) will be presented.
An example is:

```
> digi_menu
PortServer CM: Serial CM Connection menu
1 ttyS1 2 snoopy 3 linus 4 lucy
5 charlie 6 vt100-ssh
Type 'q' to quit, a valid option [1-6], or anything else to refresh :
```

selecting option 2 will telnet/ssh to the server snoopy. The names defined using the parameter serverfarm are used to make up the list. When no name is configured, ttyS<N> is used where N is the port number.

the digi_menu script has the following line options:
-p : Displays Ethernet IP Address and TCP port instead of server names

```
PortServer CM: Serial CM 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

```
PortServer CM: Serial CM 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 PortServer CM.
-h : lists script options

APPENDIX B HARDWARE SPECIFICATIONS

This chapter has all the information you need to quickly and successfully purchase or build RS-232 cabling for use with PortServer CM. It focuses on information related to the PortServer CM, but most of the information applies to any RS-232 cabling

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.

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

DTE —> RS-232 —> DCE —> communication line –> DCE —> RS-232 –> 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 9600 bps and 19200bps (used in most automation and console applications) to 115,200 bps (used by the fastest modems).

Cable Length

The total capacitance of a cable affects the integrity of transmitted data. As a rule of thumb, the total capacitance of a cable (including the connectors) should not exceed 2500 pF. Serial interface cable is usually rated in Pico Farads per foot. Therefore, if a cable has a capacitance of 50 pF/ft, and the connectors are 100 pF each, the maximum recommended cable length is 46 feet. If the cable is rated at 12.5 pF/ft, the maximum recommended cable length is 184 feet, and 5 pF/ft cable can be run up to 460 feet. In situations where low-capacitance cable (e.g. Category 5) is unavailable, or very long cable runs are required, "short-haul" modems, available from suppliers such as Black Box, can be used to increase the effective range of the RS-232 interface. Short-haul modems are similar to standard modems, except that they are connected directly to each other via a cable instead of going through a telephone circuit.

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”.

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

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). 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. They 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 Digi or a cable vendor.

To Connect To	Use Cable	Part Number
DCE DB-25 Female (standard) - Analog Modems - ISDN Terminal Adapters	Cable 1 – RJ-45 to DB-25 M straight-through (Custom) This custom cable can be ordered from Digi or other cable vendors. A sample is included with the product ("straight-through").	76000639
DTE DB-25 Male (standard) - Serial Terminals - Old PC COM ports - Most serial printers - Some Console Ports - Most automation devices	Cable 2 – RJ-45 to DB-25 F crossover (custom) This custom cable can be ordered from Digi or other cable vendors. A sample is included with the products ("Console").	76000638
DTE DB-9 Male (standard) - Newer PC COM ports - Most Mice and pointing devices - Some automation devices	Cable 3 – RJ-45 to DB-9 F crossover (custom) This custom cable can be ordered from Digi or other cable vendors. A sample is included with the products (console).	76000637
DTE RJ-45 Netra (custom) - Sun Netra Console Ports - Cisco Console Ports	Cable 4- RJ-45 to RJ-45 crossover (custom) This custom cable can be ordered from Digi or cable vendors using the provided wiring diagram.	76000636

Cable Diagrams

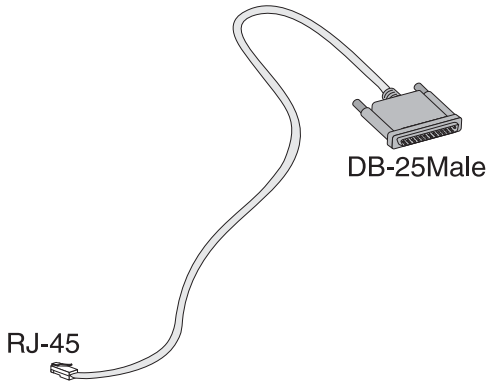
Before using the following cable diagrams refer to the tables above to select the correct cable for your application. Sometimes, crossover cables are wired slightly differently depending on the application. A “complete” crossover cable would connect the TxD with RxD, DTR with DCD/DSR, and RTS with CTS 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: Digi CM RJ-45 to DB-25 Male, Straight Through (P/N 76000639)

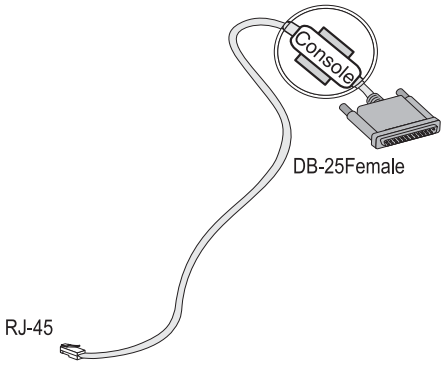
Application: It connects Digi CM products (serial ports) to modems and other DCE RS-232 devices.

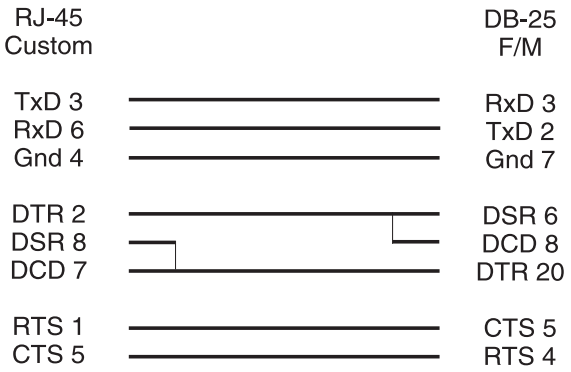


RJ-45 Male		DB-25 Male
TxD 3	———	TxD 2
RxD 6	———	RxD 3
Gnd 4	———	Gnd 7
DTR 2	———	DTR 20
DSR 8	———	DSR 6
DCD 7	———	DCD 8
RTS 1	———	RTS 4
CTS 5	———	CTS 5

Cable #2: Digi CM RJ-45 to DB-25 Female, Crossover (P/N 76000638)

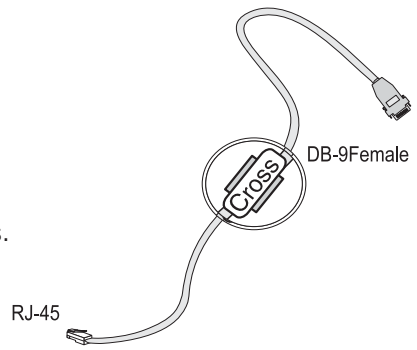
Application: It connects Digi CM products (serial ports) to console ports, terminals, printers and other DTE RS-232 devices.

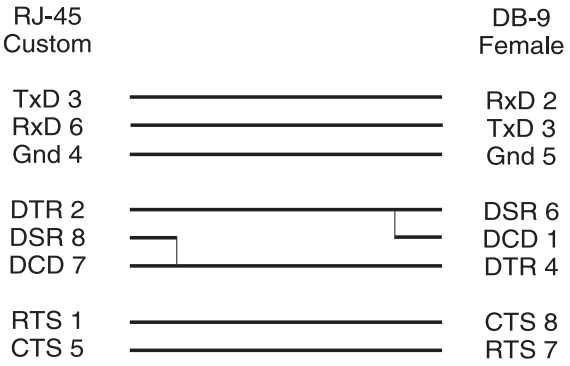




Cable #3: Digi CM RJ-45 to DB-9 Female, Crossover (P/N 76000637)

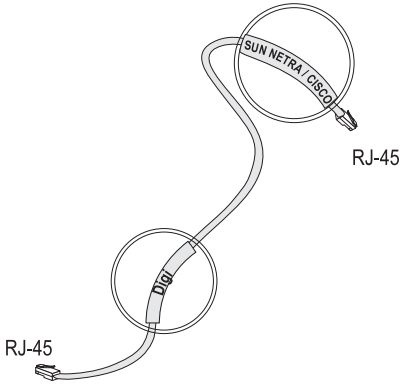
Application: It connects Digi CM products (serial ports) to console ports, terminals, printers and other DTE RS-232 devices.





Cable #4: Digi CM RJ-45 to Netra RJ-45, Crossover (P/N 76000636)

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



RJ-45 Custom		RJ-45 Netra
TxD 3	———	RxD 6
RxD 6	———	TxD 3
Gnd 4	———	Gnd 4
DTR 2	———	DSR 7
DCD 7	———	DTR 2
RTS 1	———	CTS 8
CTS 5	———	RTS 1

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 4 follow.

The Complete pslave.conf File Provided with the PortServer CM

```
#
# 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
# Eth IP address 192.169.160.10/24 (MTU=1500)
# protocol socket_server
# host IP 192.168.160.8/24
# syslogd IP 192.168.160.1
# Radius Server IP 192.168.160.3 (authentication and accounting)
# authentication radius
#

#
# Ethernet configuration.
#
# These parameters should only be configured in the file
# /etc/network/ifcfg_eth0  _IF_ the customer will not be using the
# digi_ras/portslave aplications. If the digi_ras/portslave aplications are
_NOT_
# used put all ifconfig commands for the ethernet directly in the
# /etc/network/ifcfg_eth0.
#
# The digi_ras application OVERWRITES the ifcfg_eth0 file with the
# values configured here.
```

```
#
# The PortServer CM 32 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.
#
#conf.dhcp_client       1

conf.eth_ip     192.168.161.5
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'. 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 on the machine 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
# s1.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 PortServer CM 32.
#   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-radius
#
# 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 server:  syslog is the IP address of a remote syslog daemon.
facility
#                 is a value from 0 to 7 which is sent to the syslog server to
#                 indicate in which file the syslog messages should be stored.
#                 The file /etc/syslog.conf on the syslog server contains a
mapping
#                 between facility numbers and server log files.
#
conf.syslog     192.168.160.1

#
# Syslog facility for portslave
#
conf.facility   7

#
# Syslog facility for Data Buffering
#
```

```
conf.DB_facility        7

#
# 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

#
# Syslog server message level. An integer between 0 and 7. Zero: does not send
# syslog messages to the syslog server.
#
all.syslog_level        4

#
# Syslog Console message level. An integer between 0 and 7. Zero: does not
send
# syslog messages to the console.
#
all.console_level       4

#
# Authentication type - either "local", "radius", "none", "remote"
# "local/radius", "radius/local", or "RadiusDownLocal".
#
# If the authentication type is configured as "local/radius" the portslave
# first tries to authenticate locally.  If it fails, portslave 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    radius
#
# 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
#all.authhost2  192.168.160.4
#all.accthost2  192.168.160.4
#
# The shared secret used by RADIUS.
#
all.secret      digi

#
# Default protocol.
#
# Valid values are "login", "rlogin", "telnet",
# "ssh", "ssh2", "slip", "cslip", "ppp", "ppp_only", "socket_client",
# "socket_server" and "socket_ssh".
#
#  ppp_only ==> PPP over leased lines (only authentication PAP/CHAP)
#
#  ppp      ==> PPP with terminal post dialing (Auto detect PPP)
#
all.protocol    socket_server

#
# 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\
                CM 32 - Portslave Internet Services\n\
\r\n\
     Welcome to terminal server %h port S%p \n\
\r\n\
Customer Support: http://www.digi.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 \
#               "" ATM0 \
#               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        hard

#
#  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         1

#
# PPP options - used if a PPP session is autodetected.
# Note that mru and mtu are both set to the MTU setting.
#
#all.autoppp    %i:%j novj \
#               proxyarp modem asyncmap 000A0000 \
#               noipx noccp login auth require-pap refuse-chap \
```

```
#               mtu %t mru %t \
#               ms-dns 192.168.160.5 ms-dns 0.0.0.0 \
#               plugin /usr/lib/libpsr.so

#
# PPP options - User already authenticated and service type is PPP.
#
#all.pppopt     %i:%j novj \
#               proxyarp modem asyncmap 000A0000 \
#               noipx noccp 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, the Terminal Server will send a modem status message
# to the remote device to see if the connection is still up.
#
all.poll_interval       0

#
# 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 mandadory 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.  A file (/var/run/ttyS#.data)
# is created on the PortServer CM 32 and all data received from the port is
# captured.  The file contains a maximum of 1024k, which is overwritten each
# time the maximum is reached.  This file can be viewed using the normal Unix
# tools (cat, vi, more, etc.).
```

```
#
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 recomended]
#
# all.syslog_level should be greater than or equal to 5, and data_buffering
# non-zero for syslog generation.
#
all.syslog_buffering 0

#
# Show Data buffering menu options if parameter is set to zero
# MENU:
# "A non-empty Data Buffering File was found. Choose wich action
#  should be taken ( (I)gnore, (D)isplay, (E)rase or (S)how 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 PortServer CM 32
# through one of its ports (only useful if assigned to a specific port).
# This alias will only be used if the port is 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

#
# Users that are allowed to sniff sessionsI (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

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
```

APPENDIX D CUSTOMIZATION

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

```
# 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/digi_wdt_led wdt led

# PortServer RAS
::once:/sbin/cron
::once:/sbin/snmpd
::once:/sbin/digi_buffering
::once:/sbin/digi_ras
::once:/sbin/sshd -f /etc/ssh/sshd_config
::once:/sbin/ex_ntpclient
::wait:/sbin/fwset restore
```

To customize the PortServer CM, change these lines or add others. 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 PortServer CM 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’.

Digi provides a development kit which allows changes to be made to the PortServer CM’s software. However, Digi does not provide free technical support for systems modified in this way. Any changes are the responsability of the user.

APPENDIX E THE WEB CONFIGURATION MANAGER

An HTML server to facilitate administration and configuration of the PortServer CM is supplied. To use this feature:

1. connect the PortServer CM to a terminal using the console cable,
2. log in to the PortServer CM from the terminal with root privileges,
3. change the IP address/mask/gateway using the command
   `netconfig`
   use IP numbers appropriate to your system (refer to Chapter 4 for more information on configuring your network parameters).
4. Open a browser (Netscape, Internet Explorer, etc.) and type the IP address of the CM's Ethernet interface (200.200.200.1 above). A page similar to that shown in Fig. E.1 will appear. You may also user secure socket layer by replacing `http:` with `https:` in the web URL address section of your web browser.

Type root in the username field and dbps 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.

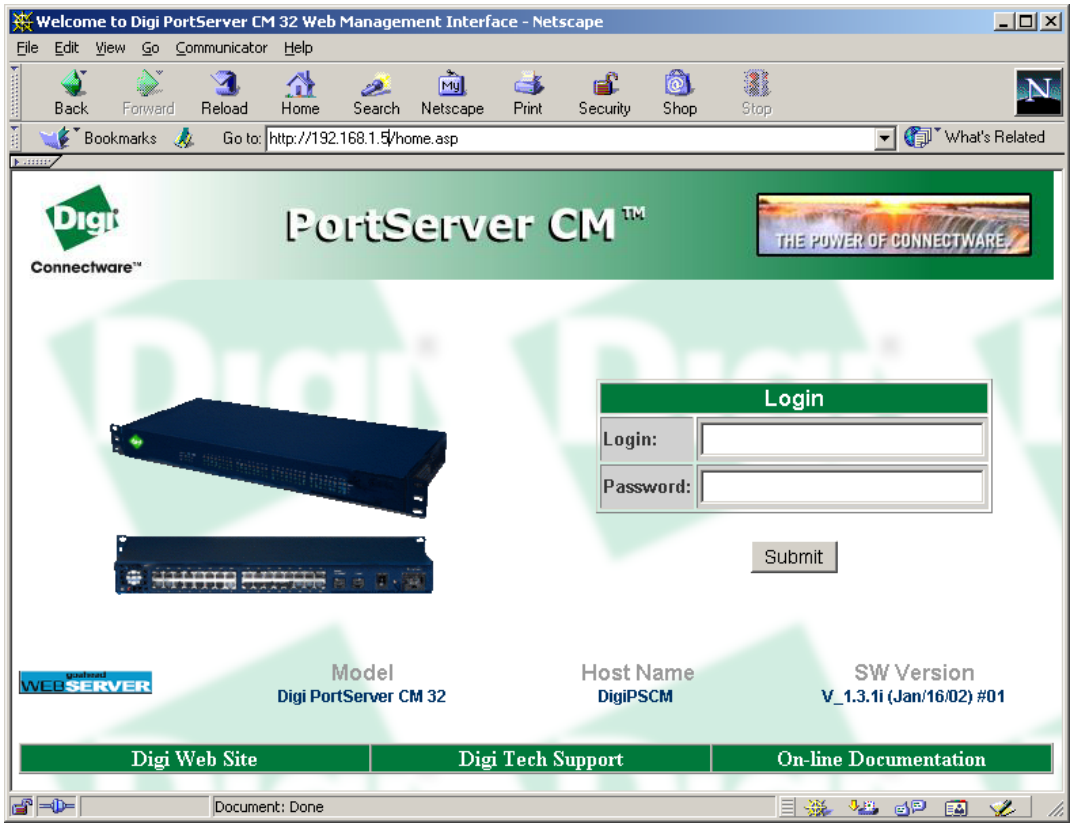


FIGURE E.1 MAIN PAGE OF THE WEB CONFIGURATION MANAGER

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. E.2

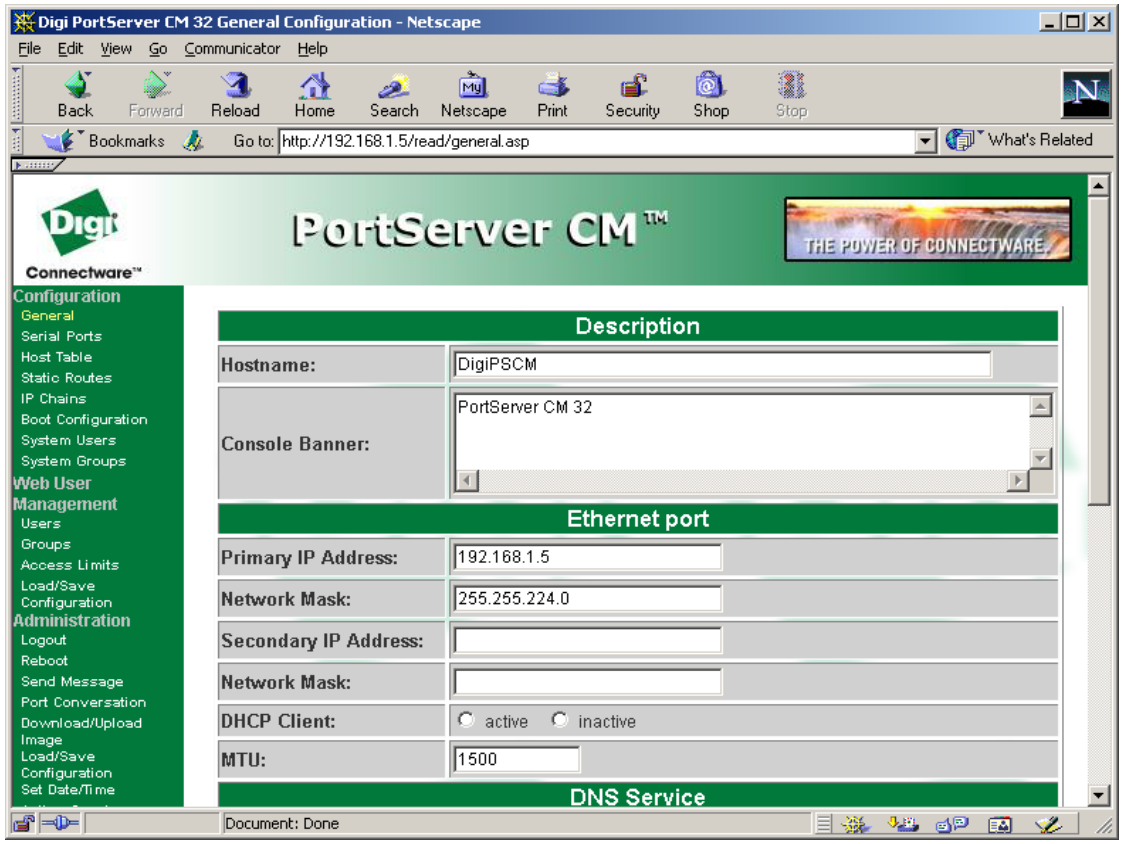


FIGURE E.2 GENERAL PAGE OF THE WEB CONFIGURATION MANAGER

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.

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

FIGURE E.3 THE CONFIGURATION SECTION

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

FIGURE E.4 THE WEB USER MANAGEMENT SECTION

Link Name	Description of Page Contents
Logout	Exits the Web Manager.
Reboot	Resets the equipment.
Send Message	Sends messages to users logged in to a serial port.
Port Conversation	Does a port conversation through a serial port.
Download/Upload Image	Uses an FTP server to load and save a kernel image.
Load/Save Configuration	Uses flash memory or an FTP server to load or save the CM's configuration.
Set Date/Time	Set the CM's date and time.
Active Sessions	Shows the active sessions and allows the administrator to kill them.
Process Status	Shows the running processes and allows the administrator to kill them.
Restart Processes	Allows the administrator to start or stop some processes.

FIGURE E.5 THE ADMINISTRATION SECTION

Link Name	Description of Page Contents
Interface Statistics	Shows statistics for all active interfaces.
Serial Ports	Shows the status of all serial ports
Routing Table	Shows the routing table and allows the administrator to add or delete routes.
ARP Table	Shows the ARP cache.
IP Chains	Shows IP Chain Entries.
IP Rules	Shows Firewall, NAT and IP Accounting rules.
IP Statistics	Shows IP protocol statistics.
ICMP Statistics	Shows ICMP protocol statistics.
TCP Statistics	Shows TCP protocol statistics.
UDP Statistics	Shows UDP protocol statistics.
RAMDisk Usage	Shows the CM File System.
System Information	Shows information about the kernel, Time, CPU and Memory.

FIGURE E.6 THE INFORMATION SECTION

Troubleshooting the Web Configuration Manager

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

Try pinging, telnetting or tracerouting to the PortServer CM 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 PortServer CM 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/dbps). Edit the file /etc/config_files and insert the reference to /etc/websum.conf.