PortServer CM User Manual

90000252_B

Copyright

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International. Digi provides this document “as is,” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

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.

Contents

Chapter 1 Introduction

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.

Safety Instructions

Use the following safety guidelines to protect yourself and 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 Technical Support.
- 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, first perform 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.

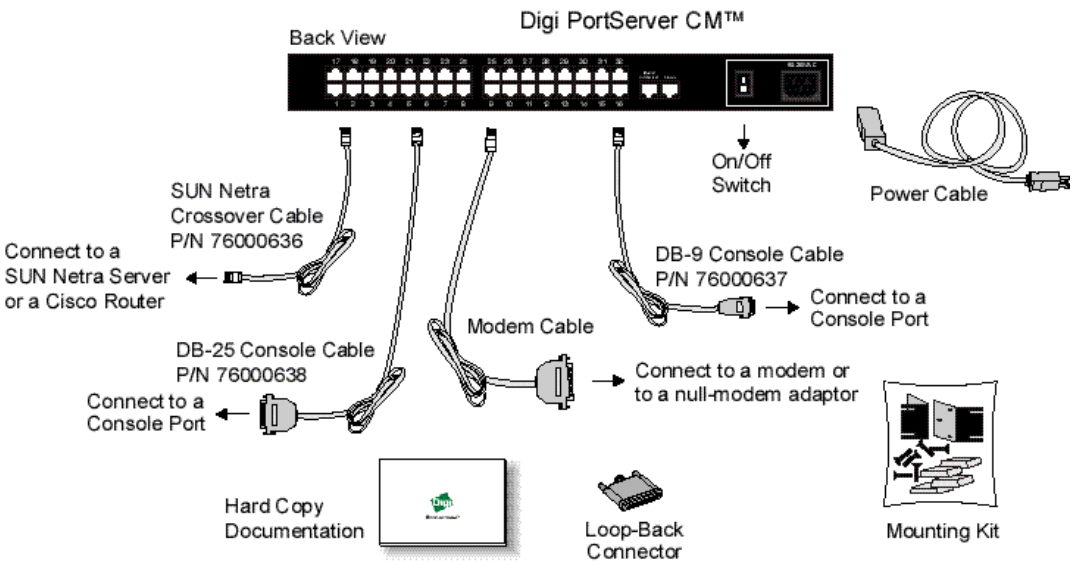
Replacing the Battery

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.

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.



LED Information

The Digi PortServer CM has a frontal array of multiple LEDs. Below is a brief explanation of what each LED represents.

Note: All 32 serial port Ready LEDs will flash when the Discover utility Locate box is selected.

LED Title	Definition
Collision	Indicates a collision on the Ethernet bus. Infrequent flashing is normal, frequent flashing indicates a saturated Ethernet bus.
Link	This LED should be on continually indicating the unit is properly terminated on the network.
CPU	This LED blinks at a rate of one second on/one second off.
100BT	This light is on if the Ethernet link is connected to other 100Base-T equipment and is working at 100 Mbps. If not, the LED will be off.
RX	Indicates the unit is receiving data. This light should be blinking continually.
TX	Indicates the unit is transmitting data.
Serial Port LEDs	
Ready	Indicates a connection to the port has been made.
RX	Indicates the unit is receiving data.
TX	Indicates the unit is transmitting data.

Summary of the Configuration Process

The PortServer CM 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 expert to configure the PortServer CM. Additional information about the files and tools needed for configuration is provided later in this manual.

Initial configuration steps are:

1. Connect the PortServer CM to the network.
2. Configure the IP settings by using the Discover Utility or the netconfig utility.
3. Edit the pslave.conf file from the command line interface or the Web Management Interface. 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.
4. Activate the changes.
5. Test the configuration to make sure the ports have been set up properly.
6. Save the changes and restarting the server application.

Chapter 2 Configuring the PortServer CM

Discover Utility

The Discover utility is a web-based Java applet that allows an administrator to quickly and easily assign IP addresses, the Netmask, and Gateway settings to the PortServer CM from any workstation on the same network as the PortServer CM. The Discover utility sends out a network broadcast and identifies responses from the PortServer CM terminal servers.

The utility is available from the Digi website at http://cm.digi.com. The utility runs locally on the system you are using, and no information about your system or network is sent over the Internet.

To use the Discover utility do the following:

1. Open a web browser and enter the following URL in the address bar:

   `http://cm.digi.com`

2. A security warning will be displayed, indicating that the applet is signed and asking if you want to install and run the Discover utility. Choose Yes.

3. Choose Discover to have the Discover utility detect the PortServer CM's on your network.

   After completing the search, a new window will open, showing a listing of the PortServer CM terminal servers found, the firmware versions, and the MAC addresses. If IP, netmask and gateway addresses have been previously defined, these addresses will also be displayed.

4. Locate the MAC address of the PortServer CM you want to configure. The MAC address is listed on a white sticker on the underside of the PortServer CM.

5. Select the IP address cell and enter the IP address you wish to assign the PortServer CM. Enter the Netmask and Gateway IP settings as needed.

6. Choose Submit to save the new IP settings.

Choose the Locate button for a visual verification of the device. The row of serial port LEDs will flash when the Locate button is selected. Press the Locate button again to stop the flashing LEDs.

Disabling the Discover Utility

The system administrator may desire to disable the Discover utility so users can not change network configuration parameters. To disable the Discover utility, the administrator needs to modify two files, submit the changes, and save the changes to the flash memory. To disable the Discover utility, use the Web Management Interface and do the following:

1. Login to the Web Management Interface with administrator rights (root).
2. From the navigation bar, choose Configuration > Edit Text File > and enter `/etc/config_files` in the Filename cell and choose Submit.
3. Scroll to the end of the list and add the following line in the text box: `/etc/inittab` then choose Submit.
4. From the navigation bar, choose Configuration > Edit Text File > enter `/etc/inittab` in the Filename cell and choose Submit.
5. Locate the following line *::once:/bin/xcelld* and replace it with the following line *# ::once:/bin/xcelld* and choose Submit. The # (number symbol) comments the line out.
6. From the navigation bar, choose Administration > Load/Save Configuration > Save to Flash.
7. Reboot the system and the Discover utility will no longer be available.

Configuration Options

After assigning an IP address with the Discover utility, users can configure their PortServer CM in two ways. The simplest way to configure most of the PortServer CM settings is to use a browser based utility called the Web Management Interface. Chapter 7 contains information on how to access and use the Web Management Interface. The other way to configure the PortServer CM is by modifying the configuration files as explained below.

Configuring Terminal Parameters

Connect a personal computer or terminal to the PortServer CM using the console cable. If you are using a personal computer, 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* and dbps as the password. The PortServer CM runs Linux. A description of the Linux file system and basic commands is given in the chapter on Linus Basics.

Configuring System Files

Modifying Linux Files

In this step, four Linux files must be modified to identify the PortServer CM and its neighbors. An alternative to editing each file is to use the *netconfig* command. Then, the boot parameters are configured. The operating system provides a scaled-down version of the *vi* editor. A description of its features is available in the chapter on Linux.

The first file is: */etc/hostname*. The only entry should be the hostname of the PortServer CM. An example is:

```
DigiPSCM
```

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

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

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

Configuration File Parameters

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 the chapter titled **Sample Pslave.conf files**.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
- s#.* parameters change the default port parameters for individual ports.

Note: An all.* parameter can be overridden by a s#.* parameter appearing later in the pslave.conf file (or vice-versa).

Modifying Pslave.conf Global Parameters

Parameter	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. An example for this value is: *200.200.200.1*
conf.eth_mask	The mask for the Ethernet network. You may use an alternative command, *netconfig*, to configure network parameters. An example for this value is: *255.255.255.0*
conf.eth_mtu	The Maximum Transmission Unit size, which determines whether or not packets should be broken up. An example for this value is: 1500
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. An example for this value is: 0

Parameter	Value for this Example
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.
conf.lockdir	The lock directory, which is */var/lock* for the PortServer CM. This file should not be changed unless the user decides to customize the operating system.
conf.syslog	The IP address of a remote syslog daemon can be provided here, if desired. An example for this value is: 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. An example for this value is: 7

Parameter	Value for this Example
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. An example for this value is: *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. An example for this value is: *4*
all.speed	The speed for all ports. An example for this value is: *9600*
all.datasize	The data size for all ports. An example for this value is: *8*
all.stopbits	The number of stop bits for all ports is 1.
all.parity	The parity for all ports is none.

Parameter	Value for this Example
all.authtype	There are several authentication 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) • RADIUSDownLocal (local authentication is tried only when the RADIUS server is down). **Note**: 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. An example for this value is: *radius*
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. An example for this value is: *200.200.200.2*

Parameter	Value for this Example
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. An example for this value is: *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. An example for this value is: *3*
all.radretries	Defines the number of times each RADIUS server is tried before another is contacted. The default, if not configured, is 5. An example for this value is: 5
all.secret	This is the shared secret necessary for communication between thePortServer CM and the RADIUS servers. An example for this value is: digi

Parameter	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 thte address exists in the host's routing table. An example for this value is: 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. An example of this value is: \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. An example for this value is: %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. An example for this value is: hard

Parameter	Value for this Example
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. An example for this value is: 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). An example for this value is: socket_server
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.). An example for this value is: 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. An example for this value is: 0

Parameter	Value for this Example
all.dont_show_DB menu	When zero, shows a menu with data buffering options when a non-empty data buffering file is found. An example for this value is: 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. An example for this value is: !joe, mark
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. An example for this value is: peter, john

Parameter	Value for this Example
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. An example for this value is: out
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. An example for this value is: 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. An example for this value is: local
s2.tty	See the s1.tty entry in this table. An example for this value is: ttyS2
s8.tty	See the s1.tty entry in this table. An example for this value is: ttyS8

Execute the command signal_ras hup to activate the changes. Now the configuration should be tested.

Testing the Configuration

After having executed the command **signal_ras hup** to activate the changes you have made to the configuration files, test the configuration by performing the following test:

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, enter the command:

   ```
   ping 200.200.200.2
   ```

   to make sure the RADIUS authentication server is reachable.
3. Verify 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. Refer to the hardware specifications for pin-out diagrams.
4. Verify that 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.
5. Verify that the computer is configured to route console data to the serial console port.
6. From a server on the LAN (not from the console), 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:

   - Ethernet IP of the PortServer CM and assigned socket port or
   - 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 chapter on troubleshooting. Continue with Updating the System Files if the configuration is successful.

Updating the System Files

To update the system with the modifications made to the files, do the following:

1. Confirm that all files that should be saved to the flash memory are contained in the */etc/config_files* folder.

   See the chapter Upgrading and Troubleshooting PortServer CM for a complete list of these files and what programs use which files.

2. Enter the command:

   ```
   saveconf
   ```

   this command reads *the /etc/config_files* file and 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.

3. Restart the digi_ras process by entering the command:

   ```
   signal_ras hup
   ```

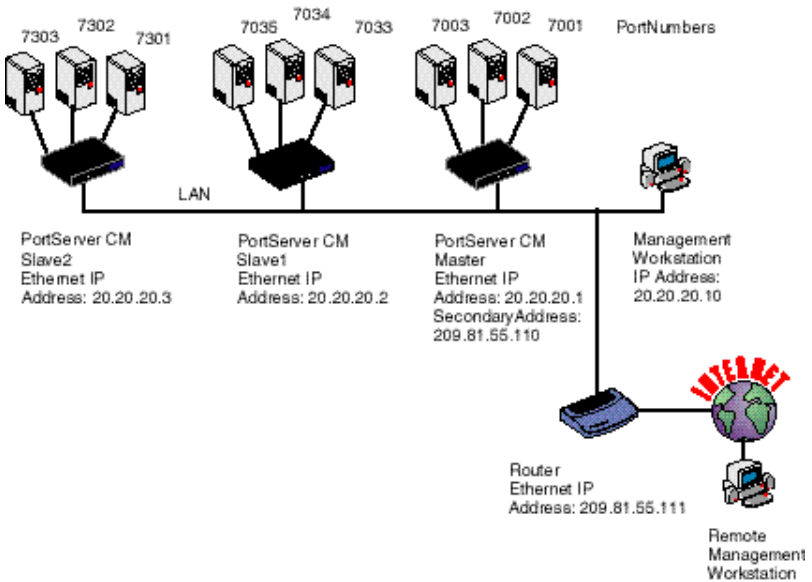
   The configuration is complete.

Clustering PortServer CM Devices

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 allows the stringing of PortServer 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 or 15 slave PortServer CMs.



Modifying Master/Slave Configuration Files

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:

Parameter	Value for this Example
conf.eth_ip	Ethernet Interface IP address. An example for this value is: *20.20.20.1*
conf.eth_ip_alias	Secondary IP address for the Ethernet Interface (needed for clustering feature). An example for this value is: *209.81.55.110*
conf.eth_mask_alias	Mask for secondary IP address above. An example for this value is: *255.255.255.0*
all.socket_port	This value applies to both the local ports and ports on slave PortServer CMs. An example for this value is: *7001+*
all.protocol	Depends on the application. An example for this value is: Socket_ssh or socket_server

Parameter	Value for this Example
all.authtype	Depends on the application. An example for this value is: 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. An example for this value is: 20.20.20.2:7033
s33.serverfarm	An alias for this port. An example for this value is: 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. An example for this value is: 0.0.0.0
s34.tty	See s33.tty. An example for this value is: 20.20.20.2:7034
s34.serverfarm	An alias for this port. An example for this value is: Server_on_slave1_serial_s2

Parameter	Value for this Example
s34.ipno	See s33.ipno. An example for this value is: 0.0.0.0
s35.tty	See s33.tty. An example for this value is: 20.20.20.2:7035
s35.serverfarm	An alias for this port. An example for this value is: Server_on_slave1_serial_s3
s35.ipno	See s33.ipno. An example for this value is: 0.0.0.0
Note: For s36-s64	use the same pattern as established in the previous examples.
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. An example for this value is: 20.20.20.3:*7301*
S65.serverfarm	An alias for this port. An example for this value is: Server_on_slave2_serial_s1
S65.ipno	See s33.ipno. An example for this value is: 0.0.0.0

Parameter	Value for this Example
S66.tty	See s65.tty. An example for this value is: 20.20.20.3:7302
S66.serverfarm	An alias for this port. An example for this value is: Server_on_slave2_serial_s2
S66.ipno	See s33.ipno. An example for this value is: 0.0.0.0
S67.tty	See s65.tty. An example for this value is: 20.20.20.3:7303
S67.serverfarm	An alias for this port. An example for this value is: Server_on_slave2_serial_s3
S67.ipno	See s33.ipno. An example for this value is: 0.0.0.0
Note: For s68-s96	use the same pattern as established in the previous examples.

The Slave PortServer CMs do not need to be configured to be 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+

To access ports from the remote management workstation, use telnet with the secondary IP address as shown below.

To access the first port of the Master PortServer CM , enter the following:

```
telnet 209.81.55.110 7001
```

To access the first port of Slave 1, enter:

```
telnet 209.81.55.110 7033
```

To access the first port of Slave 2, enter:

```
telnet 209.81.55.110 7065
```

Note: 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. To access the third port of

Slave 2, enter:

```
ssh -l <username>:Server_on_slave2_serial_s3 209.81.55.110
```

To access the fifth port of Slave 2, enter:

```
ssh -l <username>:7069 209.81.55.110
```

Setting Serial Port Buffer Size

The serial port input/output buffers have been configured at the factory for optimal performance under most circumstances, but if you find it necessary to adjust the size of the buffers, this may be accomplished by editing a text file in the /bin directory of the PortServer CM module.

Buffer sizes are defined as a number of 1 kilobyte blocks in multiples of 2048, plus 2. For example, to set the buffer sizes to 4M (1k x 4096), use the value **4098**. To set the buffer sizes to 6M (1k x 6144), use the value **6146**. Likewise, to set 2M buffers, the value to use is 2048+2 = **2050**.

Use the following procedure to set the desired buffer sizes. After you do the following procedure, go to page 6-14 pslave.conf to enable buffering.

1. Open the PortServer CM Web Management Interface and log in as the root web user. For information on using the Web Management Interface, see The Web Management Interface on page 7-1.
2. From the Configuration section, choose Edit Text File.
3. In the File Name field, enter the path of the RAMdisk configuration file:

   ```
   /bin/build_DB_ramdisk
   ```

4. In the file, locate the following two lines:

   ```
   dd if=/dev/zero of=/dev/ram bs=1k count=4098
   ```

```
mke2fs -vm0 /dev/ram 4098
```

The number, 4098, at the end of the above lines may be different, depending on your current configuration.

5. Change the number at the end of each line to the desired buffer size using the formula described above. Be sure to use the same number in both lines.
6. Be sure that you have changed only the numbers at the end of the two lines shown in Step 4. If you make an error while editing the text, choose Reset to restore the file to its last-saved state.
7. Choose Submit to save the edited file.
8. From the Administration section of the navigation bar on the left edge of your browser window, choose Load/Save Configuration.
9. Choose Save to Flash.
10. Reboot the PortServer CM either by power-cycling the device or by choosing Reboot from the Administration section of the main menu or navigation bar.

Chapter 3 Menus and Keyword Monitoring Filters

Using digi_menu to simplify port connections

Use the digi_menu script to avoid typing long telnet or ssh commands. digi_menu is ready to use immediately, and requires no configuration. 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. For more advanced menuing options, see "About Menus" on page 3-4.

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

Enter `digi_menu` with no command line options displays the default menu

```
Serial Console Server Connection menu
 1 Lucy             2 Snoopy           3 Chris            4 Ringo
 5 ttyS5            6 ttyS6            7 ttyS7            8 ttyS8
Type 'q' to quit, a valid option [1-8], or anything else to refresh:
```

Selecting option 2 will telnet/ssh to the server Snoopy. If a name is present in the serverfarm parameter for a port, that name will appear. Otherwise, tty*S**N*** is used where *N* is the port number.

The digi_menu script has the following command line options:

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

```
Serial Console Server Connection menu
 1 10.1.2.3 7001       2 10.1.2.3 7002       3 10.1.2.3 7003
 4 10.1.2.3 7004       5 10.1.2.3 7005       6 10.1.2.3 7006
Type 'q' to quit, a valid option [1-8], or anything else to refresh:
```

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

```
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   7 192.168.1.107   8 192.168.1.108
Type 'q' to quit, a valid option [1-8], or anything else to refresh:
```

-u *name*: Username to be used in ssh/telnet command. The default username is the one used to log in to the PortServer CM.

-h: lists script options

Assigning Names to Ports

Ports may be assigned names to identify their destination or purpose. The names will appear in menus instead of the generic ttyS*N* names. Use the following procedure to name ports.
Open a web browser and access the Web Management Interface by entering the name or IP address of the PortServer CM in the address bar.

1. Log in as root.
2. In the Configuration section of the navigation bar, choose Serial Ports.
3. From the Logical Ports drop down box, choose the port you wish to name, then choose Submit.
4. Enter the new name in the Server Farm parameter field (near the bottom of the page, in the SSH section) and choose Submit.
5. Repeat steps 4 and 5 for each port you wish to name.
6. In the Administration section of the navigation bar, choose Load/Save Configuration.
7. Choose Save to Flash.
8. In the Administration section of the navigation bar, choose Restart Processes.

9. Choose Stop digi_ras. After a pause, Status field should change from Active to Inactive, and the text on the button should change to Start digi_ras. If it does not, repeat this step.

10. Choose Start digi_ras.

About Menus

Introduction

The use of multi-level menus makes the connections to peripheral devices on the network or serial ports much easier for PortServer CM users. This feature enables a system administrator to define menus containing a list of actions a user or group of users can perform. The menu capability is available from the command line interface in text format. Most console management users will access the menus through Telnet or SSH, but the menus are also available from a terminal connected to a serial port.

Menus must be created by the system administrator. Menus are written in XML (eXtensible Markup Language - see XML Basics on page 3-24 for help with XML tagging), and may be created and edited with the text editor in the Web Management Interface. For administrators that do not have access to a web browser, a limited version of *vi* is included on the system to use for editing the files. A sample menu with the filename menu.xml and a default template with the filename defaultmenu.xml are included to help create new menus. They are located in the /etc/menu directory. All menus for a specific user or group of users must be contained in a single file.

A menu consists of a list of menu items. Choosing a menu item executes the action or takes the user to a sub-menu. Page Up and Page Down keys allow the user to scroll pages that are larger than screen size. Each menu item has a key that is an ASCII character, entering the key character executes the menu item or opens a submenu. The Q or q key returns the user to the parent menu. The administrator may assign any key to function as quit. A + sign beside a menu item indicates a sub-menu.

Menu Hierarchy

Global Menu

The top-level menu is always menu.xml, which is not really a menu in the purest sense (it is not interactive, and doesn’t display anything), but is used to define global parameters for the interactive menus, and to map specific users or groups of users to custom menus that have been tailored for their needs. A default menu (defaultmenu.xml) is invoked if a custom menu has not been specified for a specific user or group.

Default Menu

The default menu, defaultmenu.xml, is a generalized, basic menu that can be used to provide quick telnet access to all of the ports on the PortServer CM. The default template provided with the product is ready to use after minimal editing.

Custom Menus

Custom menus may be created for use by specific users or groups of users. Some uses for custom menus include:

- Administration - menus may be set up to contain all of the operations required to perform common administrative tasks. For example, a menu for editing configuration files might contain commands to open the files in the vi text editor, a command to activate the changes, and a command to save the changes to flash memory.
- Restricted use - a menu may be created to allow a specific user or group to access only certain ports and execute only certain commands. These users may be prevented from having command line access, so that only commands provided in the menu may be executed. See "Adding Users" on page 3-6.

Invoking Menus

Shell Users

From the command shell, simply enter `menu`. If the global menu, menu.xml, specifies a menu for the user, that menu will be automatically invoked. If no menu is specified, the default menu will be invoked.

Menu Users

When a user is assigned to the menu interface (see "Adding Users" on page 3-6), the user's menu is displayed after they login. If the global menu, menu.xml, specifies a menu for the user, that menu will be automatically invoked. If no menu is specified, the default menu will be invoked. The Menu-only user is prevented from escaping to the command line.

Adding Users

Use this procedure to set up a user so that they are required to use a menu and have no command line access:

1. Open the Web Management Interface in a browser and log in as root.
2. Choose System Users from the Configuration section of the navigation bar.

   Note: If there is already an entry for the user you wish to restrict, you must delete the entry before continuing - entries may not be edited.
3. Choose Add User.
4. Fill in the fields for user name, password and home directory as desired; in the Shell field, enter `/bin/menu` and choose Submit.
   If the Shell field is not set to `/bin/menu`, the user is presented with the command line. To assure the users are presented with a menu after they log in, fill in the Shell field with `/bin/menu`. The menu.xml file links the username to the appropriate menu.

5. To make the change permanent, choose Load/Save Configuration from the Administration section of the navigation bar, then choose Save to Flash.

Constructing Menus

There are three major categories of tags in the PortServer CM menu hierarchy:

- Menu tags, which define characteristics of an entire menu
- Menu page tags, which define characteristics of a single menu or submenu page
- Menu item tags, which define individual menu items

Menu Definition Tags

Use these tags for defining menus on a PortServer CM terminal server.

Menu Tags	Description
<menu>	indicates a complete menu (all of the following tag pairs are contained within the <menu></menu> pair)
<define>	indicates a menu definition - the <define></define> tag pair contains the entire menu definition, and is in turn wrapped by the <menu></menu> pair
<name>	menu name
<menu>	indicates a complete menu (all of the following tag pairs are contained within the <menu></menu> pair)
<comment>	area for the administrator to insert comments (not visible to user)

Menu Tags	Description
<startPage>	id of first page to be displayed
<display>	defines the number of columns to use when displaying menus. Legal values are: **Auto** - the number of columns is determined automatically to best fit the screen **1**, **2**, **3**, ... - an integer value defining the desired number of columns
<pagelist>	contains all of the defined menu pages
<page>	indicates a complete menu page (all of the following tag pairs are contained within the <page></page> pair)
<id>	defines a unique name for this menu page
<title>	title of the page as it is to be displayed on screen
<itemList>	list of all menu items on the page (all <item></item> constructs are contained within the <itemList></itemList> pair)
Menu Item Tags	**Description**
<item>	indicates a complete menu item (all of the following tag pairs are contained within the <item></item>pair)
<id>	unique id of item

Menu Tags	Description
<label>	displayed label of item
<key>	the key to press
<command>	the full connection command to execute, as if it were typed on the command line (may not be used if <page></page> is used in the same item)
<page>	page name to be linked (may not be used if <command></command> is used in the same item)
<comment>	area for the administrator to insert comments (not visible to user)
<sortby>	defines how menu items are to be sorted when displayed on screen. Legal values are: **none**: (default) do not sort, use order as defined in file **key**: sort by key (alpha-numeric) **type**: sort by type - sub-menus first, connection items last

Basic Menu Structure

Page tags are always subordinate to menu tags, and item tags are always subordinate to page tags and menu tags. The entire menu construct must be wrapped in <root> tags and preceded by an XML version tag (this is the only tag that doesn't have a corresponding closing tag). The example below shows the skeleton of a menu.

```
<?xml version="1.0"?>
<root xmlns="cm.digi.com">
  <menu>
    <define>
      <pagelist>
        <page>
          <id>page id</id>
          <itemlist>
            <item>
              <id>item id</id>
              <label>label to display</label>
              <command>command to execute</command>
              <key>key to press</key>
            </item>
            <item>
              <id>item id</id>
              <label>label to display</label>
              <command>command to execute</command>
              <key>key to press</key>
            </item>
            <item>
              <id>item id</id>
              <label>label to display</label>
              <page>id of submenu to display</page>
              <key>key to press</key>
            </item>
          </itemlist>
        </page>
        <page>
```

```
          <id>page id</id>
          <itemlist>
            <item>
              <id>item id</id>
              <label>label to display</label>
              <command>command to execute</command>
              <key>key to press</key>
            </item>
            <item>
              <id>item id</id>
              <label>label to display</label>
              <command>command to execute</command>
              <key>key to press</key>
            </item>
          </itemlist>
        </page>
      </pagelist>
    </define>
  </menu>
</root>
```

In this example, ***item id*** is a unique name for each menu item, ***label to display*** is what appears on the screen, ***command to execute*** is a complete command line to be executed when this menu item is selected, ***id of submenu to display*** points to another menu page, identified by its <id> tags, and ***key to press*** is the key that is used to select the menu item.

Sample Menus

Global Menu (menu.xml)

Here is the sample global menu that is provided with the PortServer CM. This menu may be found in the /etc/menu directory.

In this example, one user, *root*, and one group, *nobody*, are listed. Other users and groups can easily be added by creating another set of user or group tags and substituting the name and the menu that they have access to.

Note that the Global Menu does not contain the <define></define> tag pair. Instead, the tag pair <global></global> is used. This is because it is not an active menu, but a global parameter definition.

```
<?xml version="1.0"?>
<root xmlns="cm.digi.com">
<menu>
  <global>
    <version><major>1</major><minor>0</minor></version>

    <defaultMenu>defaultmenu</defaultMenu>
    <!--This menu is used if no specific criteria are met-->

    <key><quit>Q</quit></key>
    <!--define the key to exit the menu application-->

    <label>                          <!--define labels to: -->
      <quit>Quit.</quit>             <!--exit the menu application-->
      <upper>* page up *</upper>     <!--go up one page-->
      <lower>* page down *</lower>   <!--go down one page-->
      <choice>Select an option:</choice> <!--select an option-->
      <pressakey>Press any key to continue.</pressakey>
    </label>
  </global>
```

```
  <userList>
    <user>                                  <!--define user specific menus-->
      <name>root</name>
      <menu>defaultmenu</menu>
    </user>
  </userList>


  <groupList>
    <group>                                <!--define group specific menus-->
      <name>nobody</name>
    <menu>defaultMenu</menu>
    </group>
  </groupList>

</menu>
</root>
```

Default Menu (defaultmenu.xml)

Here is the sample default menu that is provided with the PortServer CM. This menu may be found in the /etc/menu directory.

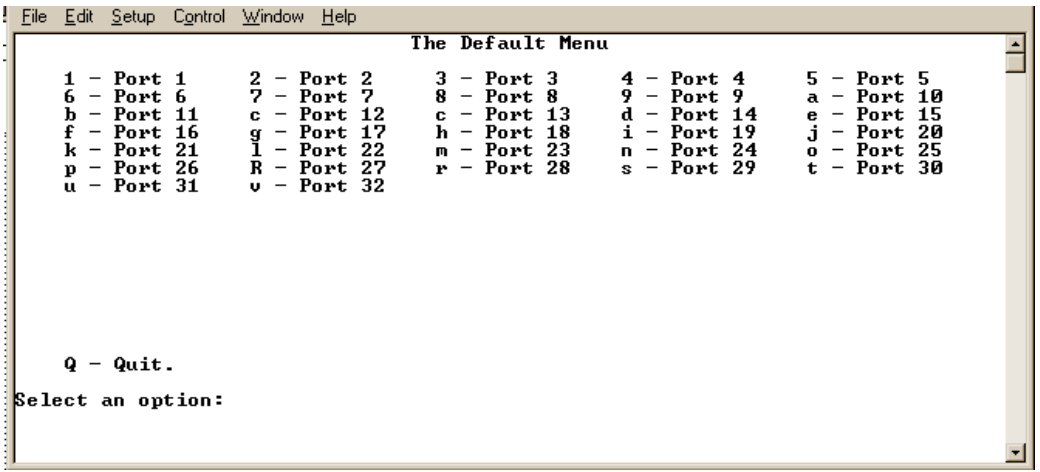
It is a fully functioning menu file and will work on any PortServer CM terminal server, you need only substitute a valid IP address. System administrators can use this file as a template and design a menu fitting their own purposes and needs. A description of the XML tags can be found at the end of this section.

Note: To save space, ports 3 through 32 have been omitted here. The defaultmenu.xml file contains the menu items for all 32 ports.

```
<?xml version="1.0"?>
<root xmlns="cm.digi.com">  <!--definition of the namespace-->
<menu>
  <define>
    <name>defaultacme</name>  <!--name/id of menu-->
    <comment>The Default Menu</comment>  <!--area for administrator to insert
comments(not visible to user)-->
    <startPage>start</startPage>  <!--first page to start-->

    <display>auto</display>  <!--display type: auto=auto OR n=the number of
column-->
    <sortby>type</sortby>  <!--key=sort by alpha key OR type=sort by type
item.Else no sort.-->

    <pageList>  <!--start a list of menus-->
      <page>  <!--start a menu-->
        <id>start</id>  <!--page unique id-->
        <title>The Default Menu</title>  <!--title displayed on screen-->
        <itemList>
          <item>  <!--define a menu entry-->
            <id>Port_1</id>  <!--unique id-->
            <label>Port 1</label>  <!--displayed text for menu entry-->
```

```
            <command>telnet 127.0.0.1 7001</command>  <!--system command to
execute-->
            <key>1</key> <!--the key to access menu entry, one char only-->
          </item>

          <item>  <!--define a menu entry-->
            <id>Port_2</id>  <!--unique id-->
            <label>Port 2</label>  <!--displayed text for menu entry-->
            <command>telnet 127.0.0.1 7002</command>  <!--system command to
execute-->
            <key>2</key> <!--the key to access menu entry, one char only-->
          </item>

<!--          Port_3 through Port_32 Omitted for Brevity        -->

        </itemList>
      </page>

    </pageList>
  </define>
</menu>
</root>
```

The following is a screenshot of how the menu would appear to the user after they have logged into the device.



Keyword Monitoring and Notification System

Digi Keyword Monitoring and Notification is an alert system designed to send notification messages to an email address, phone, or pager, using SNMP. Filters created by the system administrator allow the PortServer CM to monitor for specific keywords or phrases in a serial port's data stream. When a keyword or phrase, such as "Disk Full" or "Reboot" are detected, the PortServer CM will send an immediate alert to

the email address specified by the system administrator.

Port buffering must be enabled on the PortServer CM for the Keyword Monitoring and Notification system to work. The units are shipped with the port buffering disabled by default. The port buffering parameters can be modified in the pslave.conf files under **Data Buffering Configuration.**

Filters must be created by the system administrator. Filters are written in XML (eXtensible Markup Language - see XML Basics on page 3-24 for help with XML tagging), and may be created and edited with the text editor in the Web Management Service. For administrators that do not have access to a web browser, a limited version of *vi* is included on the system to use for editing the files.

Keyword Monitoring and Notification requires a minimum of three XML files:

- one or more **filter definition files**, which contain the words to search for and the message to send when a word or phrase is found
- one **link file**, named link.xml, which defines which ports to monitor, what filters to apply, and whom to notify in case one of the filters detects the words it is looking for - there may be only one filter link file, but it may contain many link definitions
- one **global notification file**, which contains the information required to email a notification, such as the name of the SMTP server and the addresses to put in the From and ReplyTo fields.

The PortServer CM includes samples of these files to be used as templates for creating custom monitoring and notification systems. The samples are located in the /etc/filter directory. The filenames are filter.xml (sample filter definition file), link.xml (sample link file) and notification.xml (sample global notification file).

The Filter Definition Files

A filter definition file contains the definitions for each filter.

A filter definition is bounded by the <filter><define></define></filter> tag pairs, and must include the following parameters:

Filter Name	Filter Description
<name>	name of filter
<enable>	boolean, “yes” to make filter active
<comment>	information on the filter
<grepParam>	the keyword or phrase to search for • preceding the word or phrase with -i makes the search case-insensitive • can be defined more than once to link several grep filters together
<minSendDelay>	minimal delay, in seconds, to wait before sending another alert
<sendLocalLog>	boolean, “yes” to send a syslog message
<messageTitle>	subject/title of message
<messageText>	a fixed text to include in the body of sent message
<messageMaxSize>	the maximum size, in bytes, of a message body if you want to limit the size

In the following example a filter definition file named panic.xml has been created. The filter will search for the word, “panic” If the word is discovered by the system, a message titled “Server panic” containing the text message “Digi’s PortServer wanted to let you know that one of your systems has issued a panic message” will be sent using the information in the Link and Notification files.

```
<?xml version="1.0"?>
<root xmlns="cm.digi.com">
<filter>
  <define>
    <name>panic</name>
    <comment>A filter looking for a panic string, sent by SUN servers.</comment>
    <enable>Yes</enable>
    <grepParam>-i panic</grepParam>
    <minSendDelay>60</minSendDelay>
    <sendLocalLog>Yes</sendLocalLog>
    <messageTitle>Server panic</messageTitle>
    <messageText>
      Digi's PortServer wanted to let you know
      that one of your systems has issued a panic message.
    </messageText>
  </define>
</filter>
</root>
```

The Link File

The filter link file, link.xml, defines what ports to monitor, what filters to use, and whom to notify if a filter detects a keyword. There can be only one link file, but it may contain many different links.

You may have several filter files, but only one link.xml file. If you want to use other filters or monitor other ports, you may include multiple links within the link file.

The link file uses the following parameters:

Parameter	Description
<link>	defines the parameters for a link - multiple links may be defined by including multiple <link></link> tag pairs
<snmpTrapIpAddress>	if defined, an SNMP trap will be sent to this address (you can have more than one)
<sendEmail>	boolean, yes to send an email if trigger happens
<emailTo>	if defined, an email will be sent to this recipient (multiple <emailTo> tags may be specified to notify multiple recipients)
<port>	port to which this filter is applied (multiple ports may be monitored by including a <port> tag pair for each port)

Parameter	Description
<link>	defines the parameters for a link - multiple links may be defined by including multiple <link></link> tag pairs
<snmpTrapIpAddress>	if defined, an SNMP trap will be sent to this address (you can have more than one)
<sendEmail>	boolean, yes to send an email if trigger happens
<filter>	The <filter> tag has two meanings, depending on context: When it appears inside a <link></link> tag pair, it specifies name of the filter definition to apply to the specified ports. Multiple <filter> tags may be used to monitor for several different keywords. The <filter> tag also appears just inside the <root></root> tag pair. In this context, it identifies its contents as monitoring and notification system information, as opposed to menu system information.

The sample link.xml file provided with the PortServer CM contains two links, designated by the <link></link> tags. The first link, identified in the comment field as “for my Sun boxes,” monitors port 15 for four different keywords: reboot, panic, surootfail and linkdown. If any of these words is detected in the data stream an email notification is sent to nobody@abc.com. The second link, identified as “for my Cisco routers,” monitors ports 1-7 for the word, reboot, and sends the notification to nobody@abc.com. Note that each link can have different email addresses, and that more than one email address may be specified within a single link.

```
<?xml version="1.0"?>
<root xmlns="cm.digi.com">
<filter>
  <link>
    <!--  for my sun boxes -->
    <port>15</port>
    <filter>reboot</filter>
    <filter>panic</filter>
    <filter>surootfail</filter>
    <filter>linkdown</filter>
    <emailTo>nobody@abc.com</emailTo>
  </link>
  <link>
    <!-- for my Cisco routers -->
    <port>1</port>
    <port>2</port>
    <port>3</port>
    <port>4</port>
    <port>5</port>
    <port>6</port>
    <port>7</port>
    <filter>reboot</filter>
    <emailTo>nobody@abc.com</emailTo>
  </link>
</filter>
</root>
```

The Global Notification File

The global notification file contains the information required to email a notification, such as the name of the SMTP server and the addresses to put in the From and ReplyTo fields

The notification.xml file defines the following global parameters:

<emailFrom>	the from address that will appear in the email's SMTP header
<emailReplyTo>	where to reply to the email, not always identical to the emailFrom address
<smtpServer>	the SMTP server that will handle email notifications

The sample global notification file provided with the PortServer CM defines smtp.mail.yahoo.com as the mail server, and sets cm32.abc.com as the sender (the From field of the email message). No <emailReplyTo> field is provided, so replies will be sent by default to cm32.abc.com.

```
<?xml version="1.0"?>
<root xmlns="cm.digi.com">
<filter>
  <global>
    <version>
      <major>1</major>
      <minor>0</minor>
    </version>
    <emailFrom>cm32@abc.com</emailFrom>
    <smtpServer>smtp.mail.yahoo.com</smtpServer>
  </global>
</filter>
</root>
```

XML Basics

Within a menu file are definitions of all menu info, all submenus, all connecting items and all relationships between them. Likewise, a keyword filter file contains all of the information necessary to perform keyword monitoring and notification. This information is coded in XML (eXtensible Markup Language).

Menu and filter elements are wrapped with XML tags which identify the meaning and function of the element. An XML tag consists of an identifier enclosed in angle brackets (<>). XML tags almost always occur in pairs, such that the first tag marks the beginning of an element, and the second marks the end of the element. The second tag is always the same as the first, except that it is preceded by a forward slash (/), for example, <tag>*element*</tag>.

Tag pairs may be nested within other tag pairs, such as <tag1><tag2>*element*</tag2></tag1>. Note that the entire <tag2></tag2> construct must be contained *within* the <tag1></tag1> pair, and is in fact the element for tag1. To make XML code easier to read, tagged items are often written one tag to a line, and nested elements are indented, as shown here:

```
<tag1>
  <tag2>
    element
  </tag2>
</tag1>
```

Some tag pairs, such as <menu></menu>, which defines a complete menu, always have tags nested within them, while others, such as <key></key>, which identifies a key on the keyboard, never contain nested tags.

Nested tags may have different meanings depending on context. For example, the <id></id> tag pair, when nested within an <item></item> pair, indicates the unique name of a menu item. When nested within a <page></page> pair, it indicates the name of a menu page.

Chapter 4 Linux Basics

Introduction

This chapter is designed to acquaint users with the basics of the Linux operating system. The information about Linux presented in this chapter covers the following areas:

- Adding a user
- Adding or changing a password
- Understanding the Linux filestructure
- Using *vi* editor
- Using a static routing table
- Initiating a Secure Shell Session (SSH)
- Understanding the processing table
- Using the crond utility
- Configuring DHCP
- Packet Filtering using ipchains
- Using digi_menu scripts

Changing the root Password

The PortServer CM has a single user mode 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 displaying “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 or downgrade, a second downgrade or 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.

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. Enter 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 enter "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.

Copy Command

```
cp file_name destination
```

```
cp text.txt /tmp
```

```
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.

Remove Command

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.

Make Directory Command

```
mkdir directory_name
```

```
mkdir spot
```

```
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` (print working directory) displays the name of the current directory you are in.
- 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 use-

ful 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` displays the contents of *file_name* on the screen.

The vi Editor

To edit a file using the vi editor, enter at the command prompt:

```
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.

Vi Modes and Navigation

Vi has three different modes:

- command mode for navigation within the open file. You enter command mode by pressing the <ESC> key.
- editing mode for text editing. See list of editing commands below for how to enter the editing mode.
- line mode for file saving, opening, or closing vi. From the command mode, type ":" (the colon).

When 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. The table can be displayed using one of the following commands:

`route` or `netstat –rn`.

The file /etc/network/st_routes is the PortServer CM's method for configuring static routes. See the table below. 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: cm 32
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: 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	S	/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.

Note: 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:

- crontab - the file that specifies frequency of execution, name of shell script, etc. should be set using the traditional crontab file format.
- 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 Updating the System Files on page 2-17 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 disable 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 omputer. 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

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
```

Chapter 5 Hardware Specifications

Introduction

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 “male” and the ones with holes are “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 (also known as 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

To Connect To	Use Cable	Part Number
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

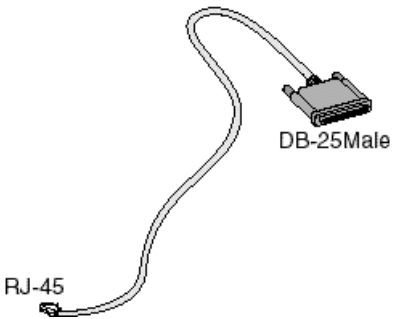
Cable Diagrams

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

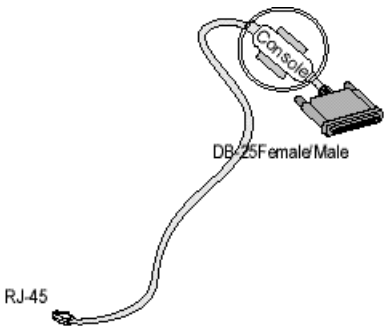
Application: It connects Digi CM products (serial ports) to modems and other DCE RS-232 devices.



RJ-45 Male		DB-25 Male
TxD3	———	TxD2
RxD6	———	RxD3
Gnd4	———	Gnd7
DTR2	———	DTR20
DSR8	———	DSR6
DCD7	———	DCD8
RTS1	———	RTS4
CTS5	———	CTS5

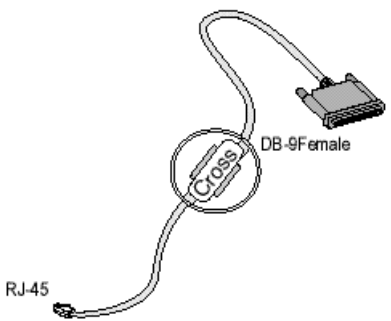
Cable #2: Digi CM RJ-45 to DB-25 Female, Crossover (P/N 76000638)

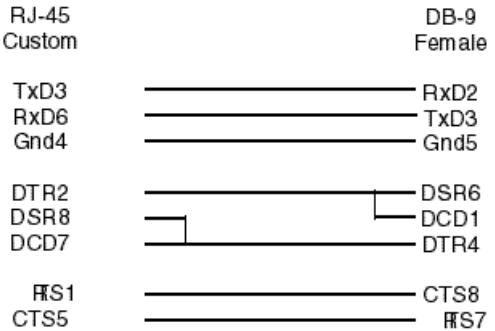
Application: It connects the PortServer CM (serial ports) to console ports, terminals, printers, and other DTE RS-232 devices.



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

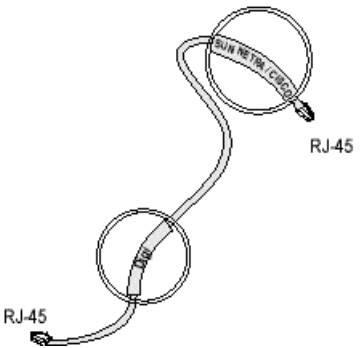
Application: It connects Digi CM products (serial ports) to console ports, terminals, printers, and other DTE RS-232 devices.





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
TxD3	———	RxD6
RxD6	———	TxD3
Gnd4	———	Gnd4
DTR2	———	DSR7
DCD7	———	DTR2
RTS1	———	CTS8
CTS5	———	RTS1

Chapter 6 Sample pslave.conf files

Sample pslave.conf File

```
# 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
# syslogd IP 192.168.160.1
# 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
# 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 can request all of its ethernet parameters to a DHCP ser.
# 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 PortServer CM User 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
# 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.
#   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 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 mapg
#               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
#
# 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
#
# 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    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
#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
#  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)
#
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\
 PortServer CM 32\n\r\n\
 Welcome to %h port %p \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        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.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 \
#               cb-script /etc/portslave/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 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 (ms), the Terminal Server will send a modem statuss
# 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 transmited 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 conection can
#                 remains 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 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. The number is equal to the
# buffer size. A file /var/run/DB/{ttyS#.data | serverfarm} is created on
# the PortServer CM and all data received from the port is captured.
# The files for all buffered ports combined can contain up to the amount
# of available memory in the ram disk. This amount can be discovered
```

```
# by typing: “df<enter>”.
# Each file is a revolving file which is overwritten as the limit of buffer
# size is reached. These files can be viewed using the normal Unix tools
# (cat, vi, more, etc.).
# If there is not enough 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 recomended]
#
# all.syslog_level should be greater than or equal to 5, and data_buffering
# non-zero for syslog generation.
#
all.syslog_buffering 0
#
# Controls the presentation of the Data buffering menu
#
# MENU:
# “A non-empty Data Buffering File was found. Choose wich 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 PortServer CM
# 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
#
# 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
```

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 responsibility of the user.

Chapter 7 The Web Management Interface

Introduction

The PortServer CM has a browser-based Web Configuration Manger which allows for easy administration and configuration. To use this feature, do the following:

1. Open a browser (Netscape, Internet Explorer, etc.) and enter the URL or IP address of the PortServer CM's Ethernet interface. A page similar to that shown below will appear. You may also user secure socket layer by replacing http: with https: in the web URL address section of your web browser.

   Note: You can find the IP address of the PortServer CM by running the Discover utility. See Discover Utility, on page 2-2.

2. Enter **`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.

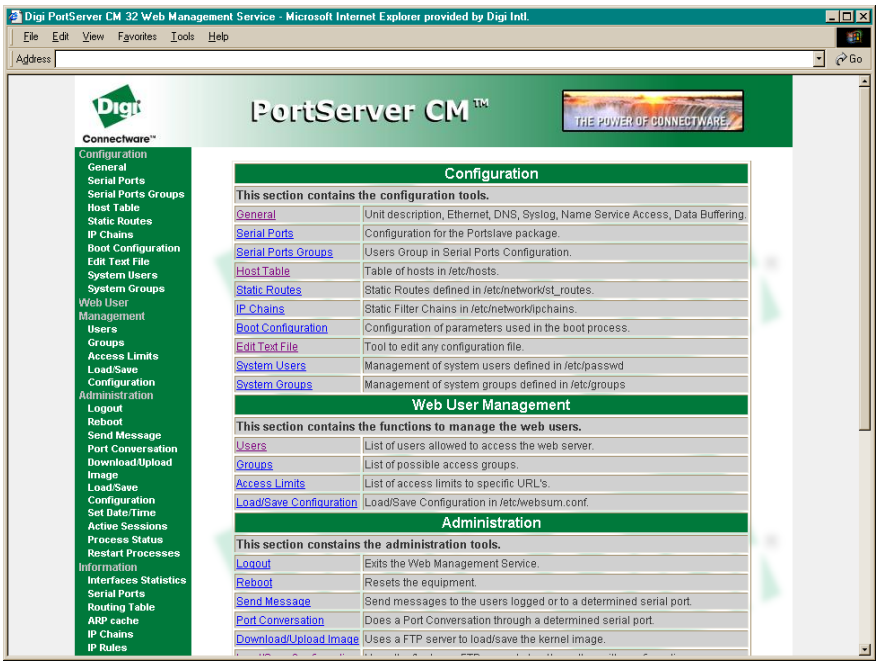
Below is the login page for the PortServer CM. The default login user is **`root`** and the password is **`dbps.`**

Changing the Password

To change the password:

1. Under Web User Management choose Users.
2. Select the radio button for the user root, then select *Change Password*.
3. Enter the new password twice and choose Submit.
4. The next page will require a new login, enter `root` and the new password.
5. From the Web User Management section, choose Load/Save Configuration > Save Configuration.
6. Next, go to Administration > Load/Save Configuration > Save to Flash.
7. To logout, choose the *Administration > Log out.*

The General Configuration page of the Web Management Interface is shown below.



Web Configuration Menus

A navigation bar is provided along the left side of the page. A summary of what each link leads to is shown in the following figures.

Configuration Section	
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.

Web User Management 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.

Administrative 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	Enables a port conversation through a serial port.
Download/Upload Image	Use an FTP server to load and save a kernel image.

Load/Save Configuration	Use flash memory or an FTP server to load or save the CM's configuration
Set Date/Time	Set the PortServer 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 adminsitrator to kill them.
Restart Processes	Allows the administrator to start or stop some processes.

Information 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 PortServer CM file system.
System Information	Shows information about the kernel, time, CPU, and memory.

Troubleshooting the Web Management Interface

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 Management Interface

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.

Chapter 8 Upgrading and Troubleshooting

Upgrading the Linux Kernel

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 entering the following command at the command prompt:

```
cat /proc/version
```

the Linux kernel version is displayed.

Troubleshooting the PortServer CM

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 earlier, 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

File	Program
/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

File	Program
/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 entering `digitest` at the command prompt. The various options are described below.

Note: 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 the chapter on hardware. 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 loopback 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.