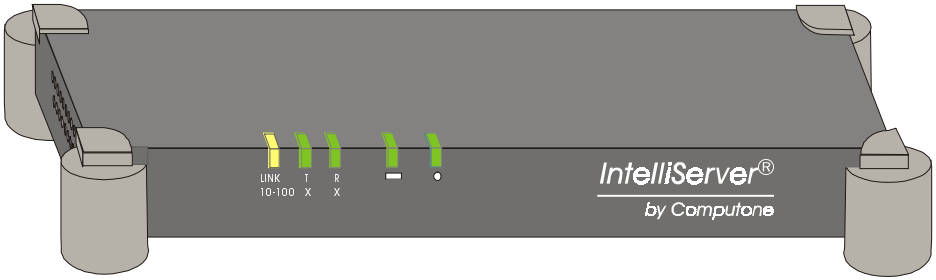
RCM 4/8 and RAS 2004/8 IntelliServer Software Guide



COMPUTONE Corporation

1060 Windward Ridge Parkway, Suite 100 Alpharetta, GA, 30005-3992 (USA)
(800) 241-3946 Outside U.S./Canada: (770) 625-0000
FAX: (770) 625-0013 email: sales@computone.com
INTERNET World Wide Web - http://www.computone.com

Computone Corporation
1060 Windward Ridge Parkway
Alpharetta, GA 30005-3992
U.S.A.

Disclaimer: Computone Corporation ("Computone") makes no representations or warranties with respect to the contents hereof, and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Computone reserves the right to revise this publication and make changes from time to time to the contents hereof, without obligation of Computone to notify any person of such revisions or changes.

FCC Statement: This equipment has been tested and found to comply with the limits of a Class A device, pursuant to Part 15 of the United States FCC regulations. 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 instruction 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 will be required to correct the interference at his own expense.

There is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception (which can be determined by turning the offending equipment off and then on), you are encouraged to try to correct or remove the interference using one or more of the following methods: (a) reorient or relocate the receiving antenna; (b) increase the separation between the equipment and the receiver; (c) connect the equipment to an outlet on a circuit different from that of the receiver; (d) consult the dealer or an experienced radio/television technician for assistance.

Industry Canada Statement: "This Class A digital apparatus meets all requirements of the Canadian Interference Causing Equipment Regulations."

"Cet appareil numérique (de la classe A) respecte toutes les exigences du Règlement sur le materiél brouilleur du Canada.

Support Information: If you require technical support, contact your Computone dealer or Computone Technical Support. The Computone Technical Support staff can be reached by phone at the following numbers, from 8:30 a.m. to 8:00 p.m. Eastern time, Monday through Friday:

(800) 241-3946 ext. 2002
(770) 625-0000 ext. 2002
(770) 625-0013 (FAX)

Technical Support can be contacted by email at the Internet address **support@computone.com**

Trademarks: Computone and IntelliServer are trademarks of Computone Corporation. All other brand names or product names are trademarks or registered trademarks of their respective corporations.

IntelliServer Software Guide P/N: 0-13116

TABLE of CONTENTS

CHAPTER 1

Starting the Web Interface

The RCM 4/8 and RAS 2004/8 provide an Internet Browser Interface to make configuration easier. To access the RCM 4/8 or RAS 2004/8, perform the following steps:

1. Enable the browser.
2. Enter the IP address of your RCM 4/8 or RAS 2004/8. For this example, we used 192.168.25.14 on a RCM 4.

NOTE: Features available differ between the RCM 4/8 and the RAS 2004/8. The RCM 4/8 has more capabilities.

The ***main menu*** (***home*** tab) is displayed.

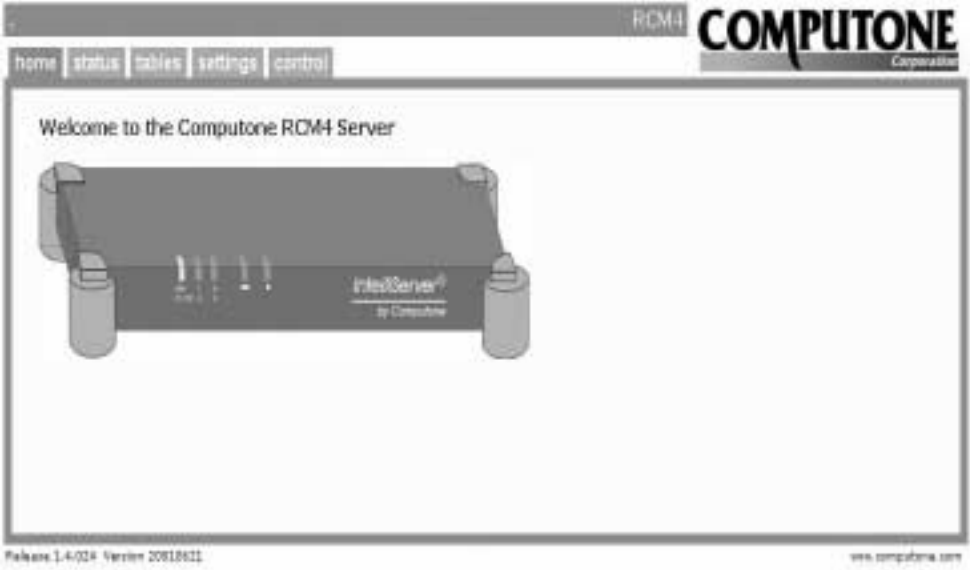


Figure 1 Main Menu (home tab)

The tabs across the top of the menu provides access to the following parameters:

Table 1 RCM 4/8 Web Interface Features

Tab	Parameters	Description
home		Welcome screen and RCM 4/8 hardware configuration information.
status		
	Activity	Shows active users on the RCM 4/8.
	Processes	Reports the status of all processes running on the RCM 4/8. A lot of this information is meaningful only to the RCM 4/8's software engineers, so details are not provided in this manual.
	Connections	Shows TCP and UDP connections and their status.
	Memory	Shows Random Access Memory (RAM) statistics.
	ARP	Address Resolution Protocol (ARP) is a protocol for determining the correct Ethernet address of a host when its IP address is known.
	Routes	Tells the RCM 4/8 where to send Internet Protocol (IP) packets based on the IP address.
tables		
	Name Servers	On larger networks, a single host (or small group of hosts, for redundancy) is given the responsibility of storing host names and addresses. Such a host is called a *nameserver* and its job is to listen for requests from other hosts and supply IP addresses for particular names.
	Gateways	The gateway table contains *static routes* which are automatically added when the RCM 4/8 starts up and when any new Serial Link Interface Protocol (SLIP) or Point to Point Protocol (PPP) links are brought up. IP uses these routes to ensure that data reaches its proper destination.
	Hosts	The RCM 4/8 uses its Host Address Table to resolve host names into IP addresses. Hosts not found in the local table are resolved through external name servers. Each host name on the left is assigned the IP address on the right. New entries to the table are available as soon as they are added.
	IP Filters	Used to add and configure IP filters.
	Ethernet	Used to view and change the Ethernet address.

Table 1 RCM 4/8 Web Interface Features

Tab	Parameters	Description
	Services	The services table can be displayed. Changes to service ports take effect when the associated process starts up. For most practical purposes this means the changes don't take effect until after the changes are saved and the RCM 4/8 rebooted.
	Users	Shows the RCM 4/8 users that are configured and provides a means to add new users or delete existing users.
	Global Connections	Used to add, configure or delete a connection to a user's configuration.
	TIP	Used to assign a name to a port.
	Console	Used to view and change the console port parameters.
	Ports	Used to configure port parameters.
	IntelliSet Profiles	Used to add or configure IntelliSet Profiles.
	IntelliPrint Profiles	Used to add, delete or configure IntelliPrint profiles.
	IntelliView Profiles	Used to add, delete or configure IntelliView screen profiles.
	Remote Profiles	Used to add, delete or configure remote profiles.
	PPP Option Profiles	Used to add, delete or configure PPP Option profiles.
	PPP Authentication Table	Used to view and change the secirity levels of users.
	VPN Profiles	Used to add and configure a remote computer to the network.
	Login Scripts	Used to add, configure or delete login scripts for remote devices.
	Dial Scripts	Used to add, configure or delete dial scripts for remote modems.
	Modem Scripts	Used to view, add, or modify modem scripts.
	COBM Manager	Used to manage out-of-band connections on any or all ports on the RCM 4/8 IntelliServer.
	SNMP	Used to add managers and agents to the Simple Network Management Protocol (SNMP).
	SNMP Community	Used to add security names to the community.
	SNMP Group	Used to add group names to the groups.
	SNMP View	Used to view names on the network.

Table 1 RCM 4/8 Web Interface Features

Tab	Parameters	Description
	SNMP Access	Used to configure name access to the network.
settings		
	System	Used to set or verify system parameters, such as host name, domain name, IP address, Ethernet address, IP filter, RIP type, login prompt, password prompt and user prompt.
	Applications	Used to enable or disable web server (htppd), secure shell (sshd) or insecure shell (telnetd).
	Boot	Used to select the boot type, host, source file and retry count for system booting.
	Syslog	Used to specify Syslog Host, Syslog Facility, and Syslog Priority.
	RADIUS Authentication	Use to configure Remote Authentication Dial-In User Service (RADIUS).
	RADIUS Accounting	Use to configure RADIUS Accounting parameters.
	RIP	Use to enable or disable and to configure Routing Information Protocol (RIP).
	Secured Shell	Use to configure the Secure Shell (sshd).
Control		
	Shutdown	Use to shutdown the RCM 4/8.
	Save to NVRAM	Use to save the current working configuration to non-volatile-RAM (NVRAM).
	Save to host: file	Use to save the current working configuration to a TFTP host.
	Scan Ports	Allows the scanning of all ports.

Home Tab

The ***home*** tab shows the hardware configuration of the RCM 4/8.

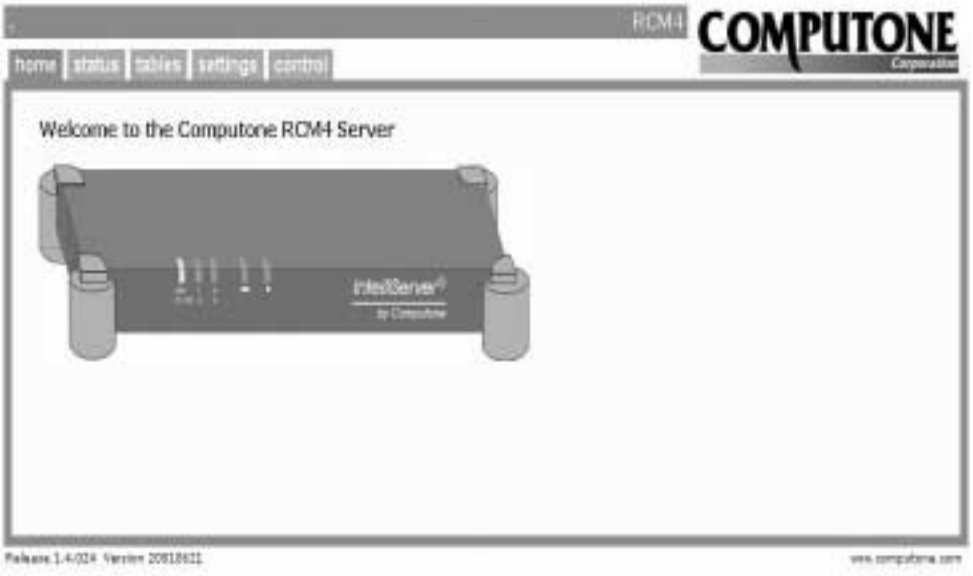


Figure 2 Home Tab Screen

The RCM 4/8 comes with the Engine Card and one 4-port or 8-port REX Card. The Engine Card is installed in the bottom slot. A single REX serial interface card is also installed in slot 1. This card has a maximum of 8 serial ports and these ports are numbered 1 - 8.

CHAPTER 2

Viewing the System Status

To understand why the system is operating in a particular manner, view the system status.

Status Tab

Selecting the ***status*** tab displays the following screen:

Figure 1 Status Tab Screen

The status parameters available for examination are:

Table 1 *Status* Parameters

Parameter	Description
Activity	Shows active users on the RCM 4/8.
Processes	Reports the status of all processes running on the RCM 4/8. A lot of this information is meaningful only to the RCM 4/8's software engineers, so details are not provided in this manual.
Connections	Shows TCP and UDP connections and their status.
Memory	Shows RAM statistics.
ARP	Address Resolution Protocol (ARP) is a protocol for determining the correct Ethernet address of a host when its IP address is known.
Routes	Tells the RCM 4/8 where to send IP packets based on the IP address.

Activity

Selecting the ***Activity*** parameter displays the following screen, which gives information on which ports are active and who is the owner.

Figure 2 Activity Screen

Processes

Selecting the ***Processes*** parameter displays the following screen:

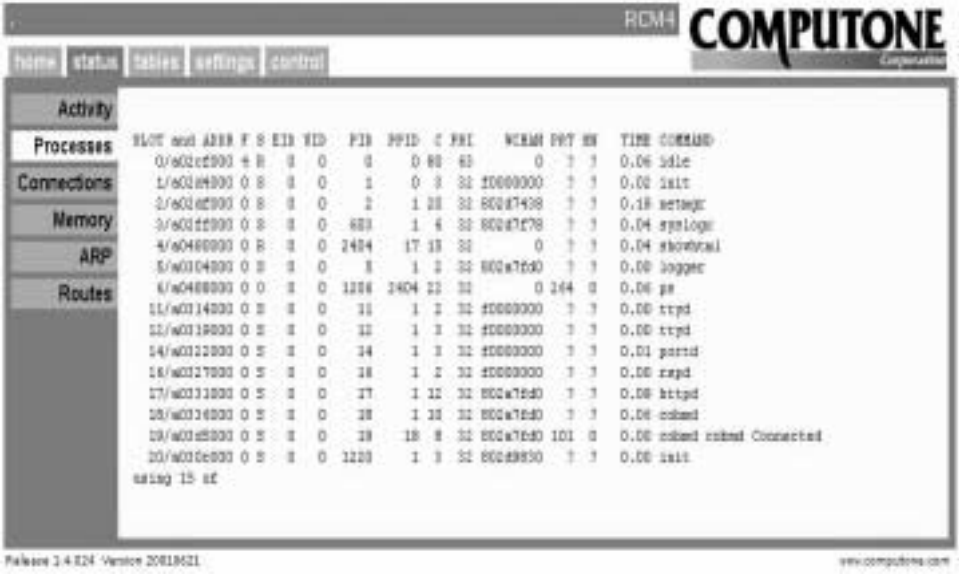


Figure 3 Processes Screen

A lot of this information is meaningful only to the Computone's software engineers, so details are not provided in this manual. The columns that are of concern are:

- **PRT** - The port number the processes is using. Port numbers 200 and 201 represent the sessions created when telneted into the RCM 4/8. A question mark under this column indicates *daemon* processes not associated with a particular port.
- **COMMAND** -The name of the process or command that is running on that port.
- **TIME** - The number of seconds of CPU time this process has used since it started.
- **PID** - The Process ID number.

Connections

Selecting the ***Connections*** parameter displays the following screen:

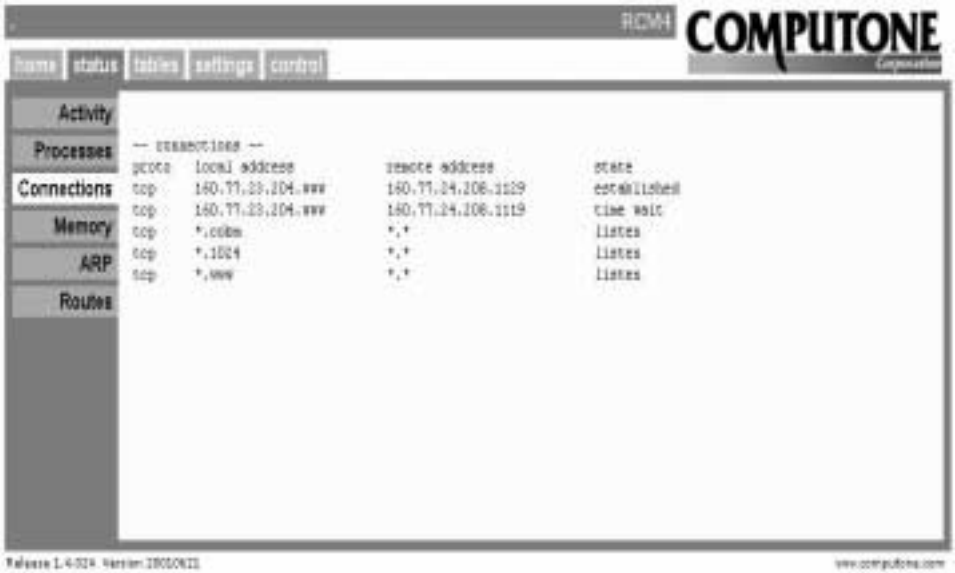


Figure 4 Connections Screen

This screen shows the active connections, their addresses and their current state.

Memory

Selecting the ***Memory*** parameter displays the following screen:

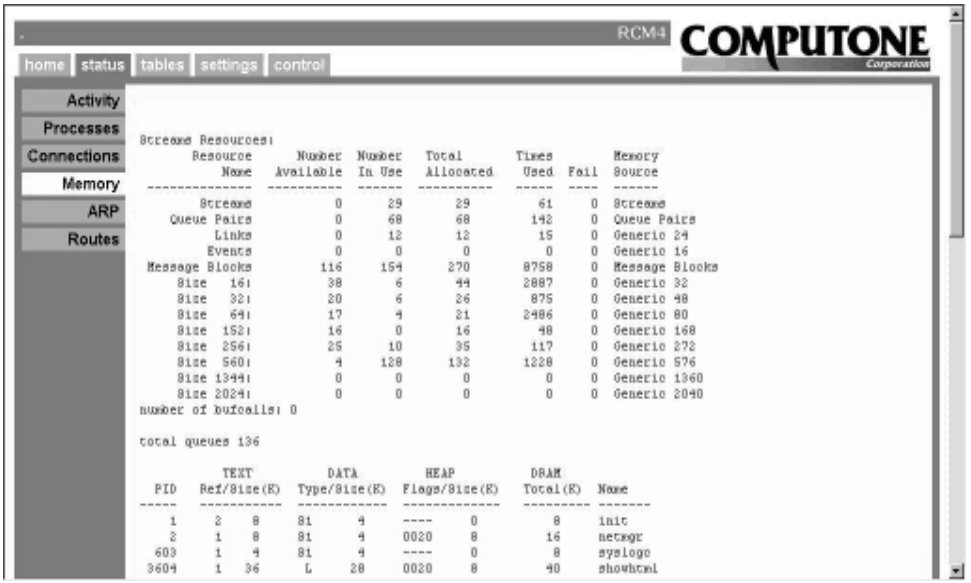


Figure 5 Memory Screen

This screen shows information about Streams Resources, Process IDs, and Kernel Resources.

Table 3 shows the information displayed for the Process IDs.

Table 2 Process IDs

PID	TEXT Ref/Size(K)	DATA Type/Size(K)	HEAP Flags/Size(K)	DRAM Total(K)	Name
1	1 8	S1 4	---- 0	8	init
2	1 8	S1 4	0020 8	16	netmgr
603	1 4	S1 4	---- 0	8	syslogc
4	1 8	S1 4	---- 0	8	telnetd
5	1 12	S1 4	0020 12	20	logger
1206	1 40	S3 12	0020 8	24	ksh
7	1 4	S1 4	---- 0	8	rshd
8	2 8	L 16	---- 0	20	ttyd
9	2 8	L 16	---- 0	20	ttyd
7810	1 8	L 24	---- 0	28	ramstat
11	1 12	L 16	0020 8	28	portd
13	1 4	S2 8	---- 0	12	rspd
14	1 16	L 24	0020 32	60	httpd

Plus DRAM for Text Segments = 68 K
Total DRAM allocated to processes = 328 K

Table 4 shows the information displayed about Kernel Memory Resources.

Table 3 Kernel Memory Resources

Kernel Memory Resources:

Resource	Name	Free	InUse	Size	Flags	MemoryAllocation
Generic	16	0	0	16	S..	0(0x4K)
Generic	24	120	50	24	S..	4096(1x4K)
MessageBlocks		16	494	24	U...	12288(3x4K)
Control		0	0	32	U..Z	0(0x4K)

ARP

Selecting the ***ARP*** parameter displays the following screen:

Figure 6 ARP Screen

The **ARP** screen shows the current ARP table. This includes the following data:

• Host name - If the RCM 4/8 can determine it from the IP address stored in the ARP table.

• IP address.

• Ethernet address, if known. - If there was no response to the RCM 4/8's ARP request for this address, the address is marked **(incomplete)**.

• Number of minutes this entry has been in the table since it was last referenced.

• Option flags (**published**, **permanent**, or **trailers) -** If an entry is marked *published*, the RCM 4/8 will respond to any ARP requests it receives for that IP address's Ethernet Address. If marked *permanent*, the entry will not expire; other entries will be removed from the table if there is no activity for that host for a long period of time. The option flag, *trailers* is reserved for future use.

Using ARP to Determine Ethernet Addresses

Address Resolution Protocol (ARP) automatically maintains tables of IP addresses and corresponding Ethernet Addresses, since maintaining these manually would be impractical. When a host wants to send a packet to some other host on the local network but does not know its Ethernet address, it broadcasts a request to everyone on the local network, saying in essence, "Does anyone know what the Ethernet Address is for IP address 192.168.25.14?" Since the question is broadcast to all the hosts on the local LAN, it should be seen by the host that is being searched for. It knows its own Ethernet address and IP address and so it sends back a reply: "192.168.25.14 can be reached via Ethernet address 80:4e:5f:ca:ff:ee".

Now that the sending host knows the Ethernet address, it can send the packet. Suppose it gets another packet for the same host. Does it start all over and send an ARP request again? That would not be wise, because each ARP request is broadcast to every host on the network. If doing this for every packet, broadcast each packet to everyone. The other hosts on the network have better things to do than read broadcast messages and throw them away. So, once the host has learned the Ethernet address for a particular IP host address, it retains the information in an *ARP Table*. It always checks its local ARP table first before sending an ARP request.

Do ARP entries stay in the ARP table forever? Generally, no. It is possible to store a permanent ARP entry in the table, but normal entries are dropped from the table if they have not been used for a long time and there is usually a way to purge entries from the table manually. This handles the case where some Ethernet card has been changed, but the IP address has stayed the same. Anyone on the network with ARP table entries made before the swapped out the card have stale information. By removing the ARP reference manually (at the command line with a `delete arp <IP address>`), there is no need to wait for the entry to expire. With the old entry gone, the host will need to perform another ARP request, and in doing so gets the new information.

Permanent ARP entries are only permanent as long as the RCM 4/8 stays up. They are not stored in NVRAM the way static routes in the *Gateway Table* are,

for example. This is generally not an issue, because usually the only permanent entries dealt with are the ones created automatically for proxy ARP on behalf of remote hosts.

Proxy ARP

Usually when an ARP request is sent to the network, the target of the request can answer for itself. “Yes, I have that IP address and here is the Ethernet address.” But sometimes, it is necessary for a *different* host to answer on its behalf: “Yes, I know who that IP address belongs to, and here is its Ethernet address.” A very useful instance of this is when a host is configured to report its *own* Ethernet address as that of the target. This is known as *Proxy ARP.*

If a host is configured for Proxy ARP and it reports its own Ethernet address as being the target host’s, it had better expect to receive packets destined for that host. When it receives the packets, it will use its own routing table to send the packets to some other interface. Proxy ARP, then, would not be used if the target host were actually connected to the local LAN. If it were so connected, it could answer ARP requests on its own behalf and receive its own packets. Nor can proxy ARP be used when the target’s host IP address is not a member of the local LAN’s network. Only host addresses that are members of the local network (as determined by the network portion of their addresses) would have been sent to the Ethernet interface in the first place.

Routes

Selecting the ***Routes*** parameter displays the following screen:

Figure 7 Routes Screen

Routing is the process of directing an IP packet to its proper destination. When there is only one network, routing is trivial and so it is easy to ignore the issue. When there are several networks and the need to route packets from one network to another, the issue can no longer be ignored. When a host has a packet to send (either one it has generated itself or one it received from the network), it could do one of the following five things:

1. Send the packet to an appropriate process running on this host, because the packet is addressed to the host itself.
2. Send the packet to a local network. This would include packets addressed to other hosts on the same Ethernet LAN, for example.
3. Send the packet to a host connected to a PPP or SLIP interface.
4. Send the packet to a *different* host on the local network or PPP/SLIP interface; that host being expected to forward it to the correct host.
5. Discard the packet because what to do with it is not known.

How does the host decide what to do? To determine how a packet should be disposed of, the host first considers whether the packet is for itself. This is easy because the host knows its own IP address (or IP addresses, when the host is on more than one network). Packets for *this* host are sent to the appropriate protocol or process to be dealt with locally.

For packets addressed elsewhere, the host uses a routing table, as shown in Figure 7 . Each entry (or *route)* in the routing table has a destination address, a gateway address, and an interface.

- If the destination address is a host address, this is a route to a specific host. A route to a specific host takes precedence over other more general routes.
- If the destination address is a network address, this route applies to any destinations with host addresses on this network.
- If the destination address is zero, this is a default route. Packets sent to destinations not otherwise accounted for are sent via this route.

CHAPTER 3

Working with Tables

Most of the configuration that can be done to the RCM 4/8 is done through the use of tables. The following table lists the parameters available for configuration through the ***Tables*** tab.

Table 1 Tables Parameters

Parameters	Description
Name Servers	On larger networks, a single host (or small group of hosts, for redundancy) is given the responsibility of storing host names and addresses. Such a host is called a *nameserver* and its job is to listen for requests from other hosts and supply IP addresses for particular names.
Gateways	The gateway table contains *static routes* which are automatically added when the RCM 4/8 starts up and when any new SLIP or PPP links are brought up. Internet Protocol (IP) uses these routes to ensure that data reaches its proper destination.
Hosts	The RCM 4/8 uses its Host Address Table to resolve host names into IP addresses. Hosts not found in the local table are resolved through external name servers. Each host name on the left is assigned the IP address on the right. New entries to the table are available as soon as they are added.
IP Filters	Used to add and configure IP filters.
Ethernet	Used to view and change the Ethernet address.
Services	The services table can be displayed. Changes to service ports take effect when the associated process starts up. For most practical purposes this means the changes don't take effect until after the changes are saved and the RCM 4/8 rebooted.
Users	Shows the RCM 4/8 users that are configured and provides a means to add new users or delete existing users.
Global Connections	Used to add, configure or delete a connection to a user's configuration.
Tip Menu	Used to assign a name to a port number.
Console	Used to view and change the console port parameters.
Ports	Used to configure port parameters.
IntelliSet Profiles	Used to add or configure IntelliSet Profiles.
IntelliPrint Profiles	Used to add, delete or configure IntelliPrint profiles.

Table 1 Tables Parameters

Parameters	Description
IntelliView Profiles	Used to add, delete or configure IntelliView screen profiles.
Remote Profiles	Used to add, delete or configure remote profiles.
PPP Option Profiles	Used to add, delete or configure PPP Option profiles.
PPP Authentication Table	Used to view and change security levels of users.
VPN Profiles	Used to add and configure a remote computer to the network.
Login Scripts	Used to add, configure or delete login scripts for remote devices.
Dial Scripts	Used to add, configure or delete dial scripts for remote modems.
Modem Scripts	Used to view, add, or modify modem scripts.
COBM Manager	Used to manage out-of band connections on any or all ports on the RCM 4/8 IntelliServer.
SNMP	Used to add managers and agents to the network.
SNMP Community	Used to add security names to the community.
SNMP Group	Used to add group names to the groups.
SNMP View	Used to view names on the network.
SNMP Access	Used to configure name access to the network.

Tables Tab

Selecting the ***Tables*** tab displays the ***Name Servers*** screen:

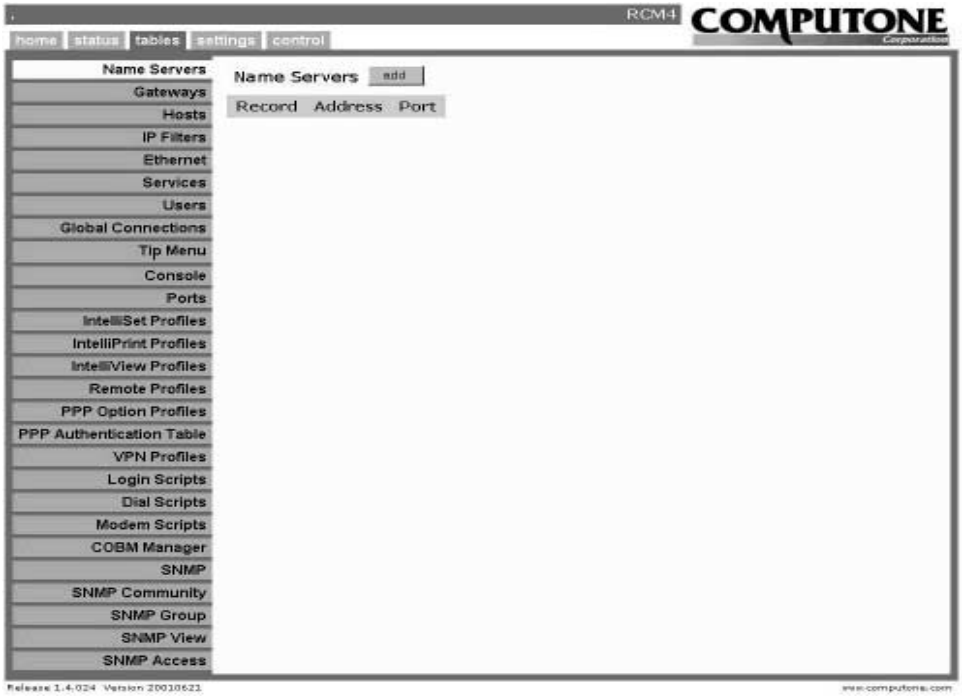


Figure 1 Name Servers Screen

On large networks, a single host (or small group of hosts, for redundancy) is given the responsibility of storing host names and addresses. Such a host is called a *name server* and its job is to listen for requests from other hosts and supply IP addresses for particular names. If the name is not found in the local table, the name server might send a request to another name server asking whether *it* might know what IP address corresponds to a particular name. The second name server might have the desired name in its table, or it might be configured to check other name servers. If an IP address is finally discovered, the name server sends it back in a reply.

The process of converting a host name to an IP address is known as *name resolution*. When names are *resolved* through an external name server, the protocol used is called *Domain Name Service* (DNS).

To add a name server, click on the ***add*** button in this screen and the following screen is displayed. Edit the ***Address*** and ***Port*** fields, then select ***Update***.

Figure 2 Adding a Name Server Screen

Gateways

Selecting the ***Gateways*** parameters displays the following screen:

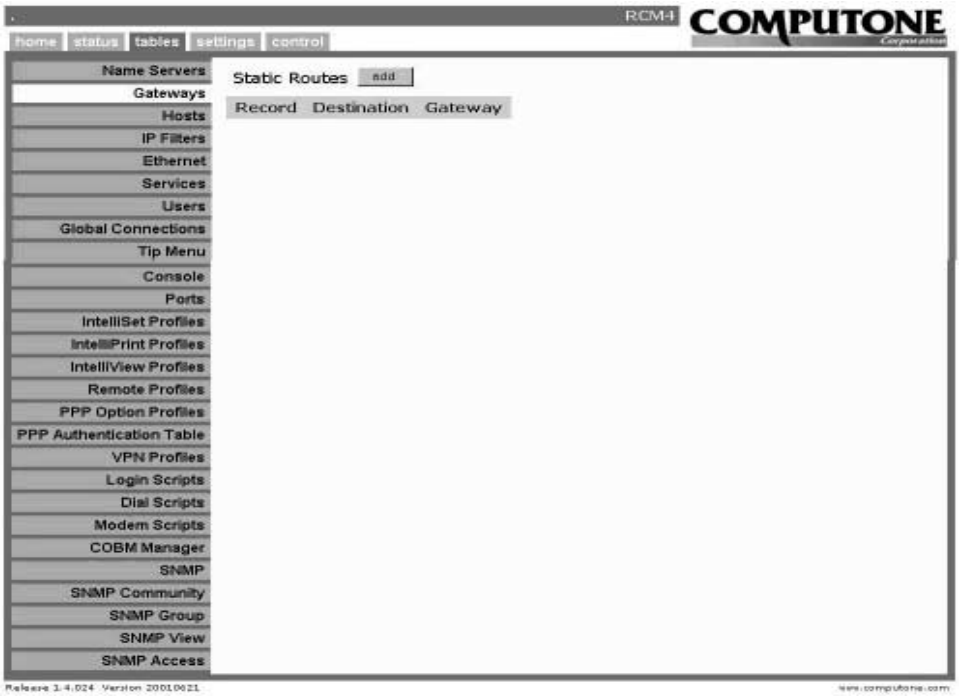


Figure 3 Gateways Screen

The gateway table shown in Figure 3 contains *static routes* which are automatically added when the RCM 4/8 starts up and when any new SLIP or PPP links are brought up. Internet Protocol (IP) uses these routes to ensure that data reaches its proper destination. This screen provides the facilities to ***Add, Delete***, and ***Copy***. After making the desired changes, select ***Update***.

Why is the gateway table reread when SLIP and PPP connections come up? There may be some routes in the gateway table whose destinations are unreachable when the RCM 4/8 is first started up, because those destinations are reached through SLIP or PPP links that are not yet up. Such a route cannot be added at that time, but *can* be added after the required SLIP or PPP link has come up.

Hosts

Selecting the ***Hosts*** parameter displays the following screen:

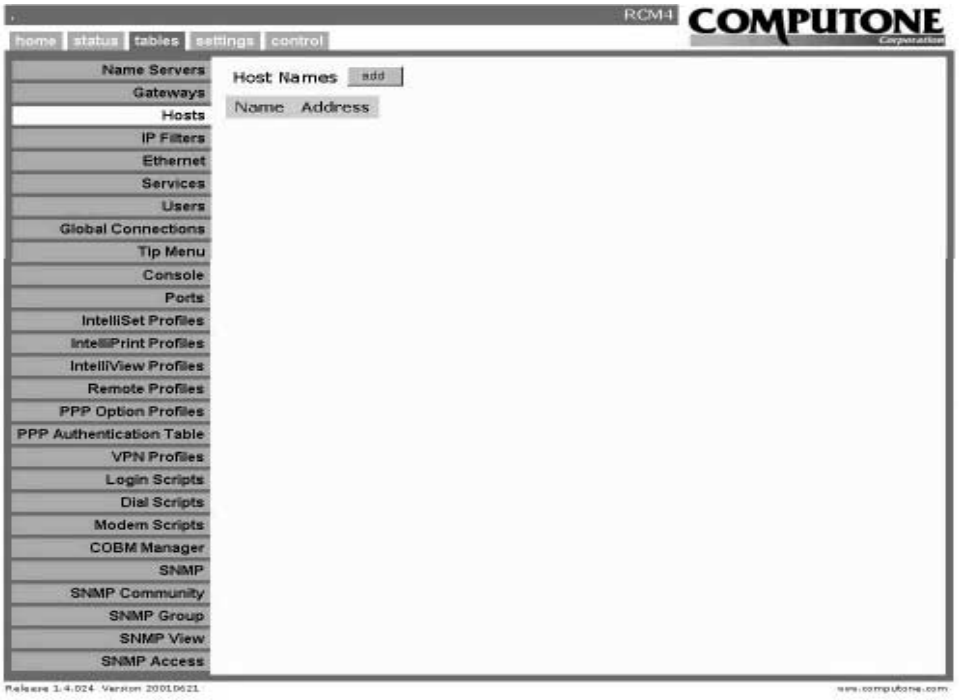


Figure 4 Hosts Screen

The RCM 4/8 uses its Host Address Table, as shown in Figure 4 , to resolve host names into IP addresses. Hosts not found in the local table are resolved through external name servers. Each host name on the left is assigned the IP address on the right. New entries to the table are available as soon as they are added.

IP Filters

Selecting the ***IP Filters*** parameter displays the following screen:

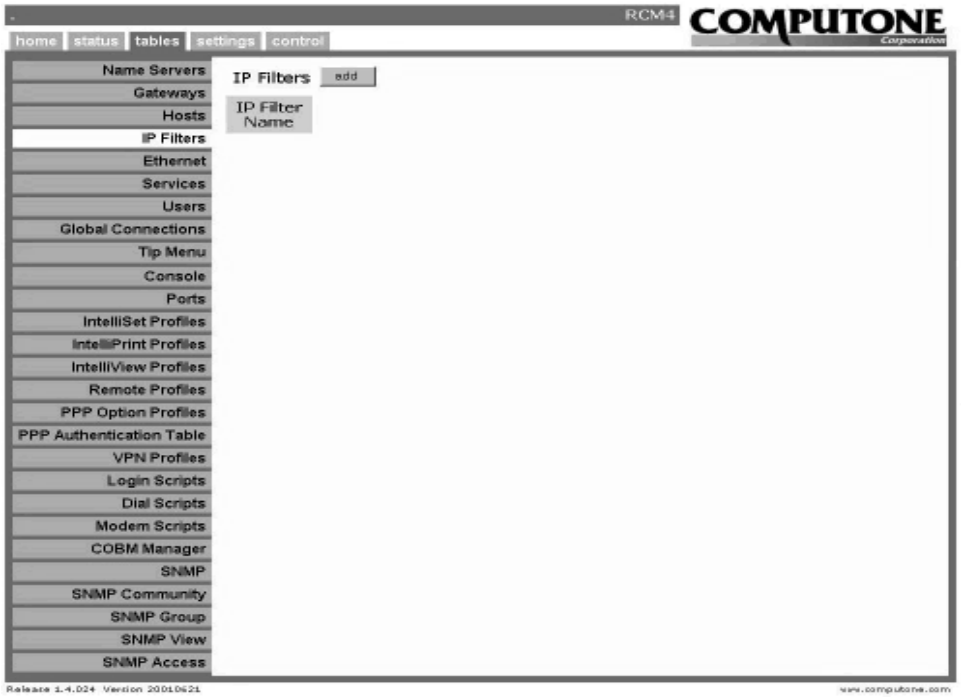


Figure 5 IP Filters Screen

To create a new IP filter, use the ***IP Filters add*** button shown in Figure 5 . When this filter is first created, it has no rules associated with it.

Once a filter has been created, add rules by clicking on the filter name, and the ***Setting the Rules*** screen appears:

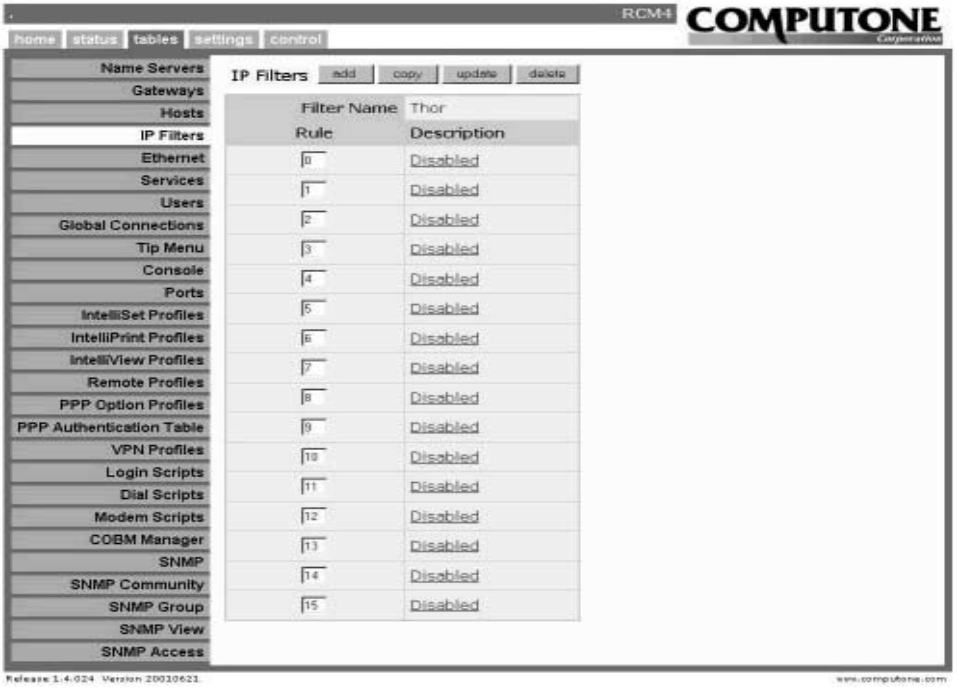


Figure 6 Setting the Rules Screen

From this screen, the rules can be set for the IP filter that was created. Click on the ***Disabled*** link beside one of the rule numbers, and the ***Setting the IP Rules*** screen appears:

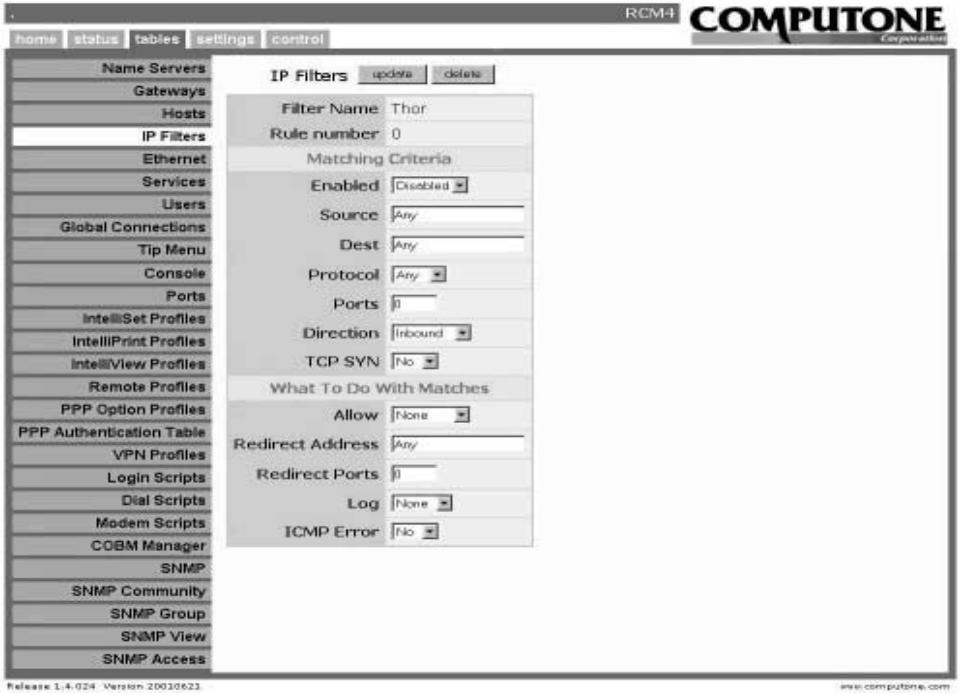


Figure 7 Setting the IP Filter Rules Screen

A rule consists of a *test* and a matching *action*. As each IP packet is filtered, the rules are applied in order. If the packet matches a rule's *test* (***Matching Criteria***), the *action* (***What To Do With Matches***) associated with that rule is performed. If the action calls for the matching packet to be allowed or denied, further testing stops. For this reason the order of rules is important. More specific tests should be specified before more general ones. If one rule defines an exception to a more general rule, the exception needs to be listed first.

There are five possible actions that can be specified.

Table 2 IP Filtering Actions

Keywords	Parameters	Description
What To Do With Matches		
Allow	None	No rule is applied.
	Require	This packet must match the test criteria.
	Allow	This packet is allowed to pass.
	Deny	This packet is not allowed to pass.
	Deny Errors	This IP packet is discarded and the RCM 4/8 sends an ICMP error message.
	Translate	Source of the outbound packets are altered.
Redirect Address		The address for the packets that match the test criteria are redirected to the address specified.
Redirect Ports		The specified redirected port.
Log	None	No statistics are kept.
	Yes/No	Do not allow or deny this packet based on the results of this test, but keep a count of how many IP packets have matched. Statistics are kept for all rules. The log action allows statistics to be kept on a condition without making an allow/deny decision based on it.
ICMP Error	Yes/No	If NO is selected, the packet is disregarded. If Yes is selected, the RCM 4/8 sends an ICMP error message back to the source.

Tests are constructed of many types of building blocks. A single test may contain several conditions which must all be true for the packet to match.

Table 3 IP Filtering Tests

Keyword	Parameters	Definition	Comments
Matching Criteria			
Enable	Disable	Enables test or disables test on packets.	
Source	*IP address* *IP address/bits*	The IP packet's source address needs to match the address in this rule.	To match IP packet addresses from a particular network, specify the number of bits to be tested after the IP address. For example, from the class B network specify the address 160.77.0.0 as 160.77.0.0/16.
Destination	*IP address* *IP address/bits*	The IP packet's destination address needs to match the address in this rule.	
Protocol	Any	Allows any packets.	
	ICMP	This must be an ICMP packet, for example, "ping".	If a service port or range of ports is specified in this rule. TCP and UDP ports in that range are matched. If no service ports are specified, all ports are included.
	GGP	This must be GGP packet.	
	TCP	This must be a TCP packet.	
	EGP	This must be a EGP packet.	
	PUP	This must ba a PUP packet.	
	UDP	This must be a UDP packet	
	IDP	This must be a IDP packet.	
Ports	Singular port number	The destination port in the IP header must match this one.	Since the destination is about network headers, port refers to the TCP or UDP service port associated with a connection; it has nothing to do with serial ports. When the port keyword is used, TCP or UDP must be specified, since the same port numbers could apply to either.
	Range (e.g. 1-35)	The destination port in the IP header must match this range.	
Direction	Inbound	The test is applied to inbound packets.	Either in or out must be specified. Specify both if the test is to apply to all packets.
	Outbound	The test is applied to outbound packets.	
TCP SYN	*Yes/No*	This matches any TCP packet that has the SYN flag set. This flag is always set in the first packet sent over a TCP connection, so this test could be included in a rule to prevent certain new TCP connections from being started up.	

Figure 8 and Figure 9 are examples of setting filter rules. In the first example this rule allows all incoming packets destined for port 21 (used for FTP connections) of host 160.77.99.30. Specifically, this allows an outsider to establish an FTP connection to one particular host

.

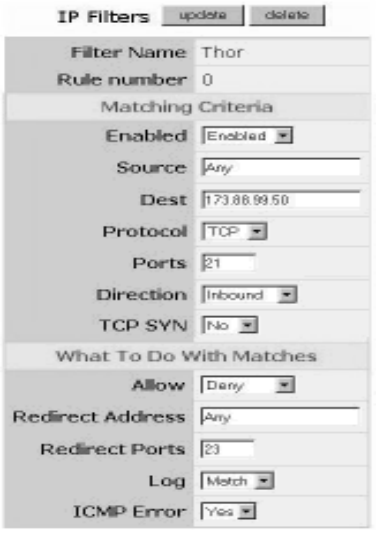


Figure 8 Example 1 of Setting a Rule

The example rule shown in Figure 21 forbids any incoming packets from host addresses in the range 160.77.128.1 - 160.77.255.254.

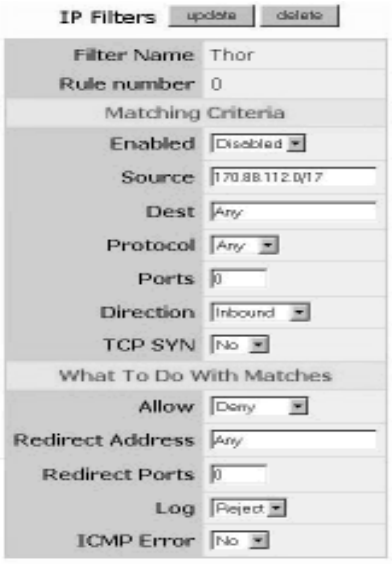


Figure 9 Example 2 of Setting a Rule

Ethernet

Selecting the ***Ethernet*** parameter displays the following screen:

Figure 10 Ethernet Screen

The ***Ethernet*** *screen* shows the ***Interface Name***, ***IP Address***, and ***IP Filter*** applied.

Services

Selecting the ***Services*** parameter displays the following screen:

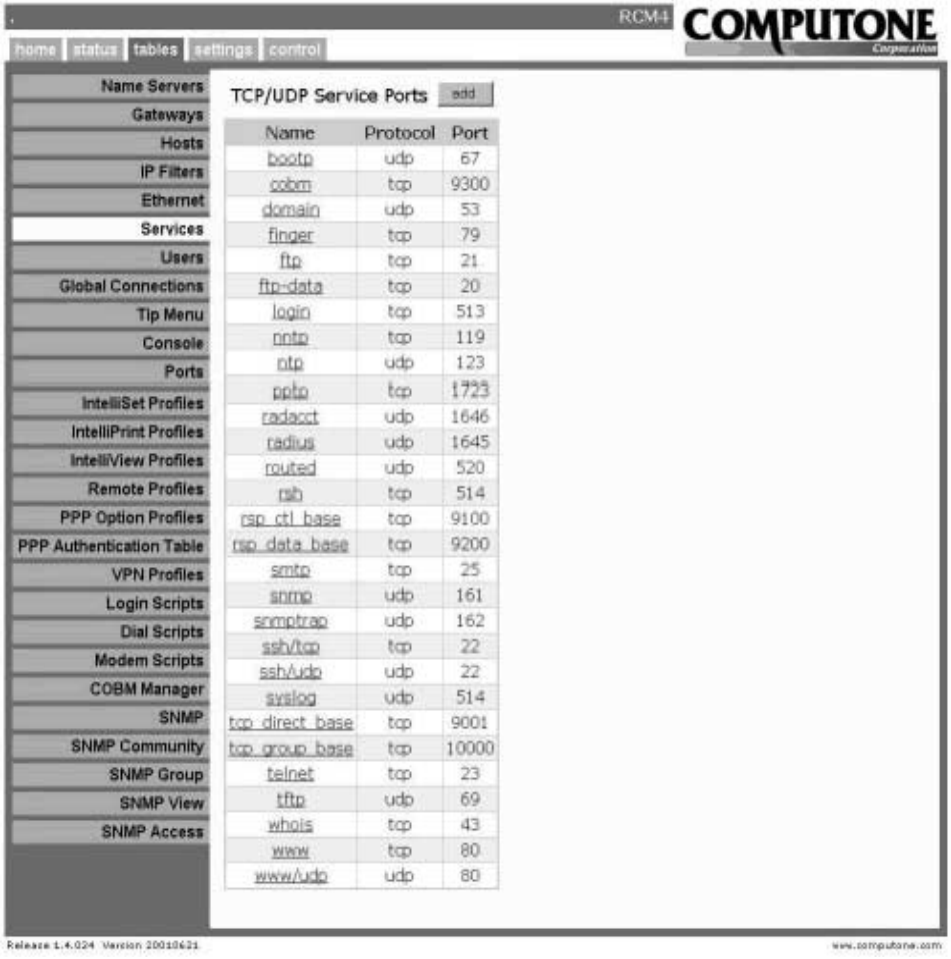


Figure 11 Services Screen

When client and server processes communicate with each other using Internet Protocol, IP addresses in the *IP header* are used to ensure that the data is sent to the proper host computer. The *IP header* also contains source and destination *port numbers*, which serve to identify which *particular* client or server on a host is the source or destination of that data.

When talking about the RCM 4/8, “ports” usually refer to *serial ports*, but these port numbers have nothing to do with serial ports; they are just numbers used in Internet Protocol. Processes which provide standard services listen on particular *well-known ports*. Client processes which want to get a particular type of service from a host try to make a connection to that *well-known port*. After it does, the server process can assign the client a different port number that applies to that particular session between those particular processes. Standard well-known port numbers have been assigned to standard services and are listed in the RCM 4/8’s *Service Ports* table. The need to change the entries will probably never occur, unless the network is extremely unusual, but the table is provided nonetheless.

To display and modify the *service ports* table, click the ***Record*** number shown in the table in Figure 11 . Several protocols are listed, each with its own *well-known port*. The column marked “Protocol” shows whether TCP or UDP protocol is used for that service. Look at the first entry in the table in Figure 11 : “telnet, port 23, tcp”. This means that if the RCM 4/8 wants to telnet into some host, it needs to contact TCP port 23 on that host. This is a multi-page table and only one page is illustrated here. *The services table may contain entries for protocols that the RCM 4/8 does not support.*

Users

Selecting the ***Users*** parameter displays the following screen:

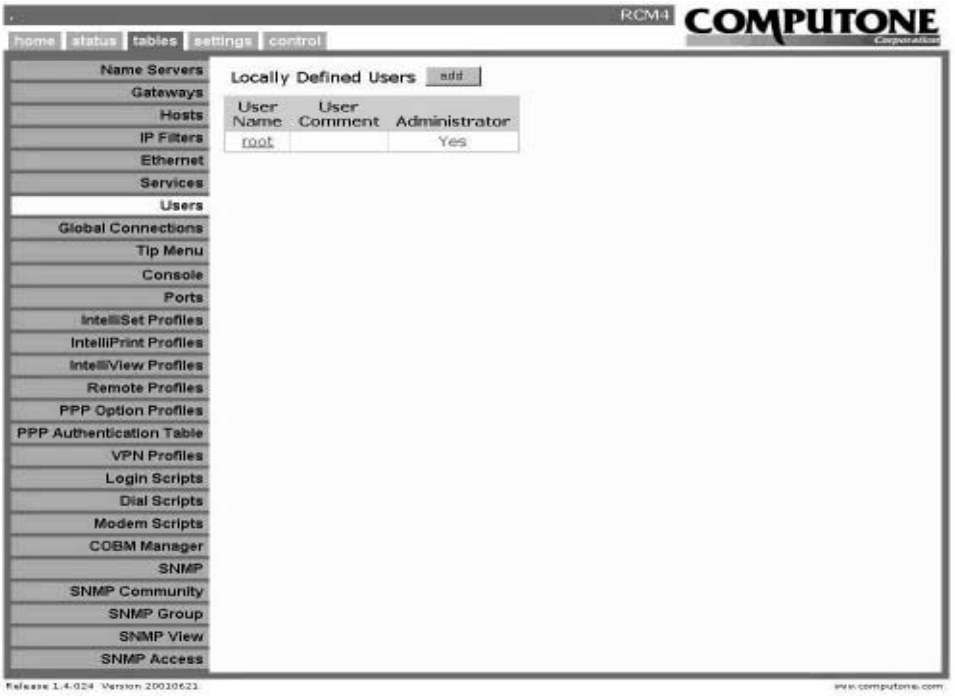


Figure 12 Users Screen

A user that is configured on the RCM 4/8 and stored locally in its *non-volatile RAM* is called a ***NVRAM user***. This is the easiest way to configure a small number of users on a single RCM 4/8. The local NVRAM is limited to storing about a hundred users, to support a larger number, they must be stored on another host. A user whose information is stored on another host on the network is known as a ***RADIUS user*** *(Remote Authentication Dial-In User Service)*, because the RADIUS protocol allows the user information to be sent from this host to the RCM 4/8. Click on a user name, and the ***User Configuration*** screen is displayed.

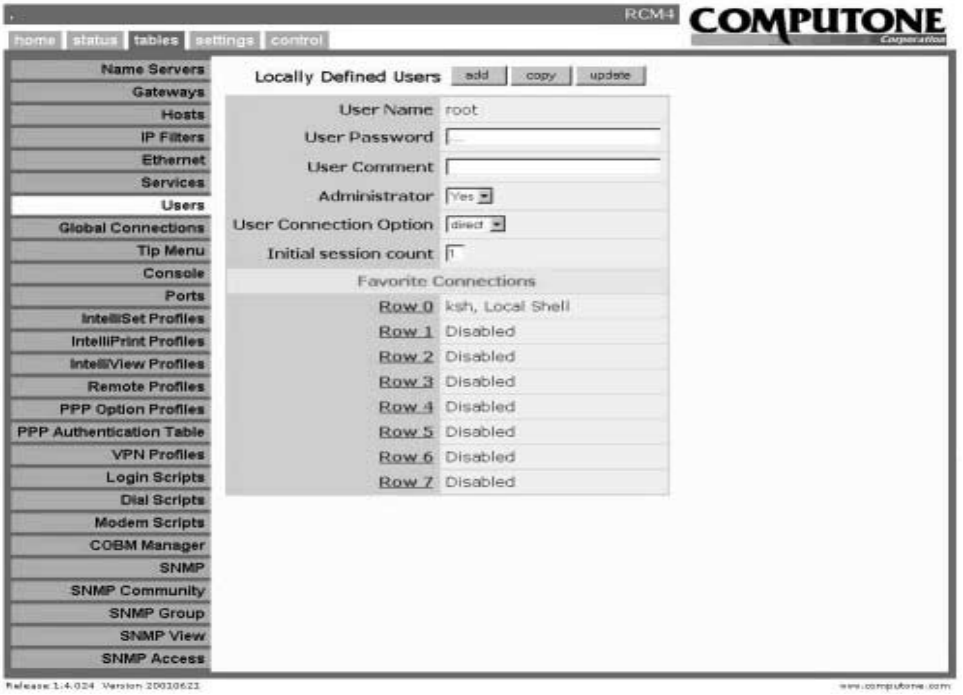


Figure 13 User Configuration Screen

Table 4 Users Configuration Entries

Keywords	Description
User Name	User account being defined.
User Password	Enter a password for this user, if one is desired.
User Comment	Enter something meaningful to identify this user.
Administrator	Select Yes or No, depending on whether this user is to have Administrator rights.

Table 4 Users Configuration Entries

Keywords	Description
User Connection Option	This is the master control for what happens when the user logs in. Direct Connect per Screen (***direct***), Selected Connection Menu (***select***), and Full Connection Menu (***full***) are used to support login users or to establish telnet and rlogin connections with other hosts on the network. Inbound SLIP (***SLIP***), Inbound CSLIP (***CSLIP***), Inbound PPP (***PPP***) are used to support dial-in users that want to establish PPP, SLIP, and CSLIP connections to computers or networks at their sites.
Initial Session Count	This applies to users that have been configured as Direct Connect per Screen and limits the number of sessions that are initially started after the user logs in.
Favorite Connections	
Up to 8 global connections can be assigned for each user that is assigned on the ***Users*** menu. To assign a global connection, first find the global connection number from the global connection table for the service wanted to use, and then unlock the row number that is going to be assigned. Then, enter the ***gc*** number in the **ROW** submenu.	

Whenever a new selected connection is added to a user's configuration, the entry is automatically added to the *global connection table*. This is a master table that contains all the connections configured for all users. Entries in the global connection table may be directly added, modified, and deleted, without working through user configuration. In most installations with lots of login users, there tends to be more users than there are places to go. If the global connection table number is known for a particular connection, then the user can be configured more quickly. More importantly, if some system-wide parameter changes, a single change is more likely to be made and affect all appropriate users. For example, perhaps lots of users are configured to rlogin to a certain host in order to perform a specific function. But later, this function is moved to a different host on the network. Each user *could* be changed separately or do the following:

- Look at the user configuration form for one of these users. In its selected connection table is the global connection number of that connection. Remember it.
- In the global connection menu, find the entry and change it. All other users using that entry are updated as well.

This is possible because the RCM 4/8 automatically forces users with identical connections to share a single global connection entry. Remember, entries must be completely identical. Even the spacing must be identical or separate entries are created. If two users were configured with identical connections and a change

was wanted for only one user, the change would be made using the user configuration. This would automatically create a new entry in the global table for the user's new connection.

Global Connections

Selecting the ***Global Connections*** parameter displays the following screen:

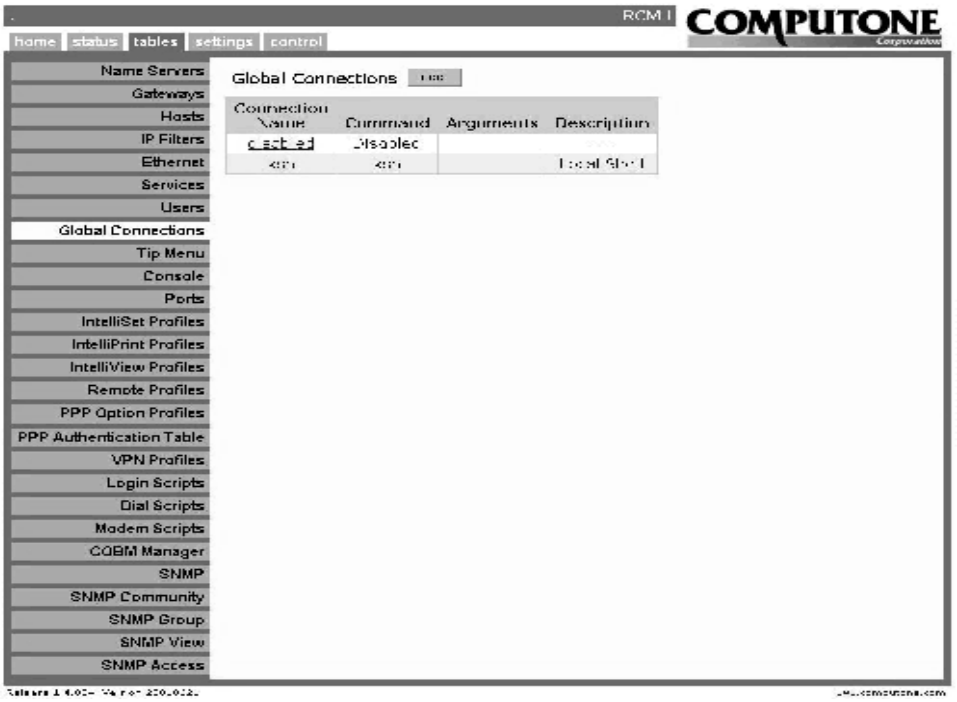


Figure 14 Global Connections Screen

Whenever a new selected connection is added to a user's configuration, the entry is automatically added to the ***Global Connection*** *table*. This is a master table that contains all the connections configured for all users. When using the global connection menu, any of the connections in this table can be run.

Entries in the global connection table can be added, modified, and deleted directly, without working through user configuration. In most installations with lots of login users, there tends to be more users than there are places to go.

If the global connection table number is known for a particular connection, then the user can be configured quickly. More importantly, if some system-wide parameter changes, the ability to make a single change and affect all appropriate users is more likely. For example, perhaps lots of users are configured to rlogin to a certain host in order to perform a specific function. But later, this function is moved to a different host on the network.

Each user *could be* changed separately, or do the following:

- Look at the user configuration form for one of these users. In the user's selected connection table will be the global connection number of that connection. Remember it.
- In the global connection menu, find the entry and change it. All other users using that entry will be updated as well. This is possible because the RCM 4/8 automatically forces users with identical connections to share a single global connection entry. Remember, entries must be completely identical. Even the spacing must be identical or separate entries are created.

If two users were configured with identical connections, and a change for one user only was desired, the change would have been made using the user configuration. This would automatically create a new entry in the global table for the user's new connection.

To change an entry in the global connection table, click on connection name and the ***Global Connections Configuration*** screen is displayed.

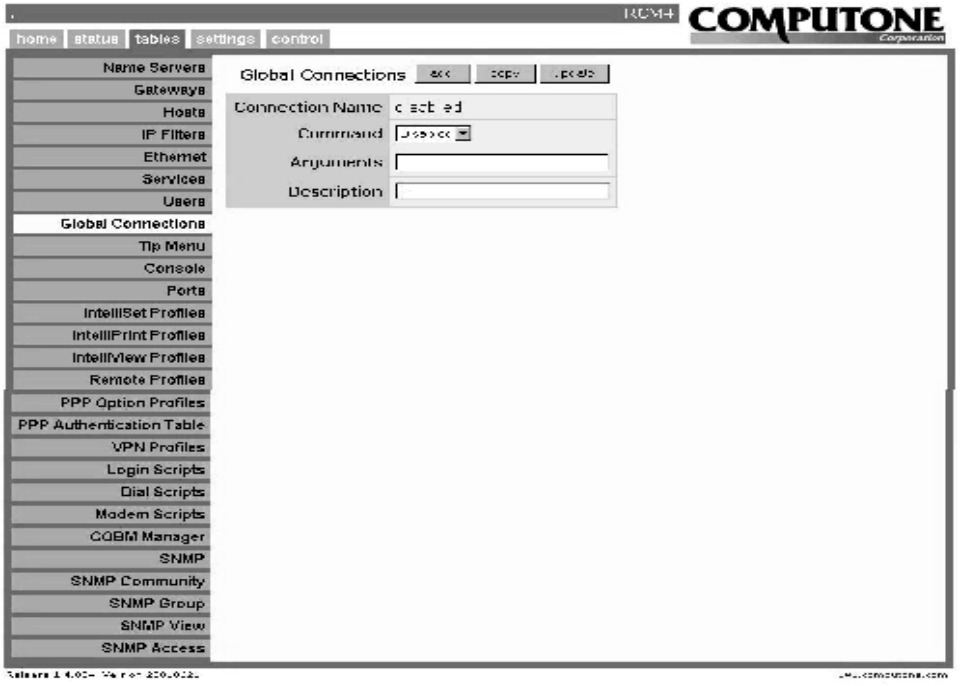


Figure 15 Global Connections Configuration Screen

The commands are pick-lists, the choices are:

- ***Free*** - entry is available to store a new global connection.
- ***Disabled*** - entry is not available to store a new global connection.
- ***shell*** - use to administer and maintain RCM 4/8 or to start a connection.
- ***menu*** - use to administer and maintain RCM 4/8 or to start a connection.
- ***rlogin*** - use to start a connection to a specified host.
- ***telnet*** - use to start a connection to a specified host.

To delete an existing global connection, try to set its command to ***Free***. If there is a user configured with this as one of his selected connections, the RCM 4/8 won't allow the change to be made. When modifying one of these entries, it affects all users whose selected connection table contains the entry.

The ***Arguments*** option provides a place to enter any arguments that are necessary for the selected command.

TIP

Selecting the ***TIP*** parameter displays the following screen**:**

Allows the user to assign a name to a port.

Console

Selecting the ***Console*** parameter displays the following screen:

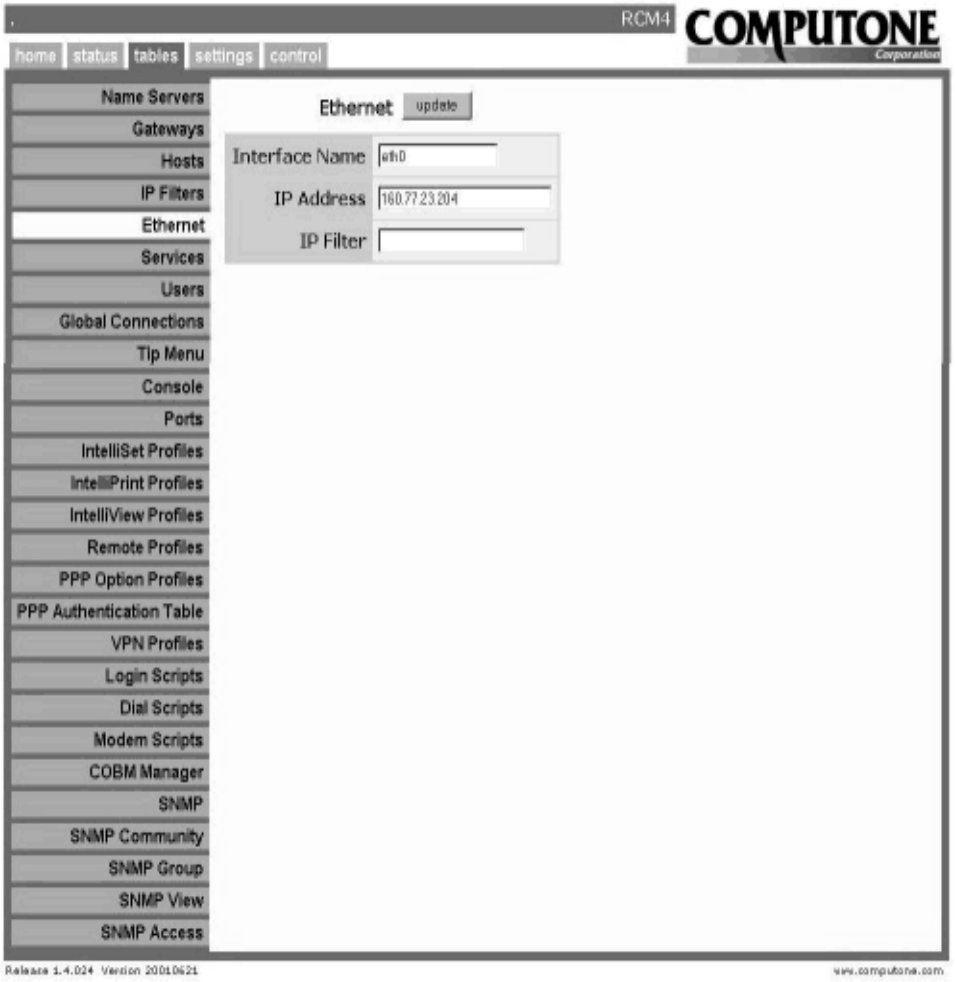


Figure 16 Console Screen

The parameter selections are defined as follows:

Table 5 Console Port Configuration Parameters

Keyword	Parameters	Definition
Console		Number of port being configured. In this case, port 0.
Type		
	Disabled	Nothing happens on this port, except the use of the commands such as **tip** and **output** to send data to it in order to test the port or configure a modem or other device.
	Login by Port, Wait	With this selection, the port sends a login prompt to the attached terminal or modem. When the user logs in, the RCM 4/8 starts up whatever connections have been configured for that user.
	Login by Virtual Screen	Generally use this setting only if the port is configured to support multiple sessions through IntelliView. Each virtual screen is sent its own login prompt, and the user must log into each virtual screen separately. When a session is ended, a new login prompt for that virtual screen is sent, but DTR is not dropped if there is any other session on this port still active.
	Auto-Login/Wait	This is almost identical to *Auto-Login*, except that instead of launching the connection immediately, the port first sends a prompt: **Press <enter> to continue**, and when the operator does this, *then* the connection is launched. This is designed to solve the quandary that occurs when a port configured as *Auto-Login* is attached to a local terminal that is always on but frequently unattended. The user logs off and walks away, and the RCM 4/8 immediately launches the connection. Suppose that connection is an attempt to rlogin to some host machine. So that machine prompts for a password. Since there is no one present to enter a password, the connection soon times out and is restarted, and times out and is restarted and so on. If the port is configured as *Auto-Login*, *wait*, then the RCM 4/8 remains at the "**Press <enter> to continue**" prompt until someone does this and the retries and time outs are avoided.
	Auto-Login	To configure a port so that the RCM 4/8 automatically starts a connection without prompting for a login, perform the following step. If the port is configured for *Auto-Login*, a user name for this port (*myname*, in the above example) must also be specified.The port would behave exactly as a *Login-by-Port* but instead of sending a login prompt, he assumes that the specified user has successfully logged in, and starts up his connections accordingly. When the session is over, the RCM 4/8 will (after waiting for *carrier* when appropriate) restart the sessions again.
	Printer	This configuration is similar to *Reverse-TCP*, except that a port configured as a *printer* can also accept connections from **rcp** and **rsh cat** clients on the network.
	Reverse-TCP	When a port is configured as *Reverse-TCP*, the RCM 4/8 accepts a TCP connection from some other host on the network. Data received from that host is sent out the serial port, and data received from the serial port is sent to the host. This is a common method of supporting printers and other "non-login" serial devices.

Table 5 Console Port Configuration Parameters

Keyword	Parameters	Definition
Type	Out-bound PPP or SLIP Connection	This configuration supports outbound PPP/SLIP/CSLIP links. The RCM 4/8 brings these links up automatically when it tries to route a packet to a network that it knows to be on the other side of one of these links. Note that this port type supports only *dial-out* connections. To support clients who are dialing into the RCM 4/8, configure the port as *Login-by-Port*.
	Login by Port/TCP	This is a combination of *Login-by-Port* and *Reverse-TCP*, and is designed to support bidirectional operation of a modem. The port must be configured as *modem enabled*, because the RCM 4/8 uses *carrier* (DCD) to sense incoming calls and determine whether there has been a disconnection. When the port is idle and there is no incoming call, the RCM 4/8 accepts TCP connections for this port from hosts on the net, just like *Reverse-TCP*. If a connection is established, the client can access the modem, send dialing commands, and connect to other systems. Anyone trying to dial in gets a busy signal because the modem is off-hook. If an incoming call comes in first, the port sends out a login prompt, like *Login-by-Port*, and as long as the incoming call is connected, the RCM 4/8 refuses or defers TCP connections from the network for that port.
	Remote Serial Port	When a port is configured as a remote serial port, it can establish communications with another host's driver. Once a host's driver establishes communication with a RCM 4/8 PowerRack, it attaches to one or more remote serial ports (RSP) and presents them as "logical" serial ports to the host operating system. In the case of NT, these appear as COM ports.
User Name		Name of port user.
Group		There are 16 groups of ports, numbered 0 to 15. Any port can belong to any group, or to no group at all. When something tries to start a reverse-TCP connection to the RCM 4/8, it can specify a particular port or a particular port group. When a port group is specified, the first available port in the group is used. A port group number can also be specified in a *Remote Profile* for an outbound PPP/SLIP/CSLIP interface. A port is configured as *Reverse-TCP* or *Login-by-Port/TCP* cannot be a member of the same group as a port configured as *Printer* or which uses *IntelliPrint*. This is because the first types suppress output processing, while the others perform it. If both types were members of the same group, the results might depend on which printer happened to be available.
Wait for CD (Modem)	Yes/No	The modem port waits for the modem to assert carrier (**DCD**), if the port was configured as a *modem port*. When a *modem port* is connected to a local terminal, the port's **DCD** is usually wired to the terminal's **DTR** or **RTS** (whichever is *not* being used for flow-control). In that case, the RCM 4/8 would be waiting not for an incoming call, but for someone to turn the terminal on. If this is a non-modem port, it is assumed that carrier is present. The RCM 4/8 waits for 1 second before continuing, after carrier (**DCD**) is detected by a *modem port*.

Table 5 Console Port Configuration Parameters

Keyword	Parameters	Definition
Wait for Input	Yes/No	This allows any attached modem or device to stabilize, before an attempt to send data to it. For example, there are some modems which assert carrier before coming out of command mode. Data intended for transmission to the remote modem may be interpreted as a command. After this 1 second delay, any data that might have been received so far is flushed before the preamble or login prompt is sent.
Speed	Selections of speed settings.	This sets the line speed at which data is transmitted and received. In addition, custom rates can be defined by setting up an IntelliSet profile and assigning that profile to a port. By using IntelliSet, a *split baud-rate* can also be specified where the port transmits at one speed, and receives at another. When line speeds and other parameters are defined using IntelliSet, those values override the ones chosen here. 50 150 1200 3600 19200 64000 230.4k 75 200 1800 4800 38400 76800 307.2k 110 300 2000 7200 56000 115200 460.8k 134.5 600 2400 9600 57600 153.6k 921.6k
Data Bits	Selections are 8, 7, 6, and 5.	How many data bits per character.
Parity	Selections are odd, even, space, mark, and none.	This controls the parity bit sent with each characters.
Stop Bits	Selections are 1, 1.5, or 2 bits.	This controls the number of stop bits that are transmitted after each character. One stop bit is generally sufficient except when connecting to devices that are very old, very slow, or very unusual. This has no effect on the receiver. One stop bit is always sufficient.
Input Flow Control	Selections are none, XOFF, RTS, and XOFF & RTS.	Since the RCM 4/8 can receive data from a serial device, it must be configured to signal when its buffers start to fill up so that the serial device stops sending data for a while.
Input Translations	Selections are CR to NL (carriage return to newline), none (no translations), and NL to CR (newline to carriage return)	Any input processing specified here affects the port's operation when it is accepting line-based input, such as at the command prompt, or when in telnet command mode. At other times, the individual applications (telnet, rlogin, menu, etc.) force the input processing to an appropriate setting. .
Output Flow Control	Selections are XON, None, XANY, CTS, XON & CTS, and XANY & CTS.	Sometimes the serial port on the RCM 4/8 is the sender, and it must avoid overrunning the terminal, modem, or printer to which it is attached. This is called *output flow control*. The *output flow control* chosen needs to match the *input flow control* of the device that data is being sent to, and vice-versa.

Table 5 Console Port Configuration Parameters

Keyword	Parameters	Definition
Output Translations	The selections are NL to CR+NL, None, CR to NL, Strip CR, CR | NL to CR+NL, and NL to CR+NL.	Any output processing specified here affects the operation of the port *only* when it is configured as a printer.
Output Expand Tabs	Yes/No	With this setting, the port will translate ascii **tab** characters to a sequence of spaces sufficient to achieve tab stops at 8-character intervals. This tab setting corresponds to the traditional tab processing performed on UNIX systems and is useful when printing output from a UNIX system using tools that expect this processing to be performed "downstream". If this parameter is set to *No* (or *disabled* using the command), then **tab** characters are sent unchanged.
INTR Character	^C	This defines the *interrupt key.* Use this key to quickly terminate commands before they have finished. In this example **^c** represents *control-c*.
ERASE Character	^H	This defines the character used to backspace a single character and erase it.
QUIT Character	^\	This defines the quit character.
KILL Character	^U	This defines the kill character.
EOF Character	^D	This defines the character used to denote the end of file character.

Ports

Selecting the ***Ports*** parameter displays the following screen:

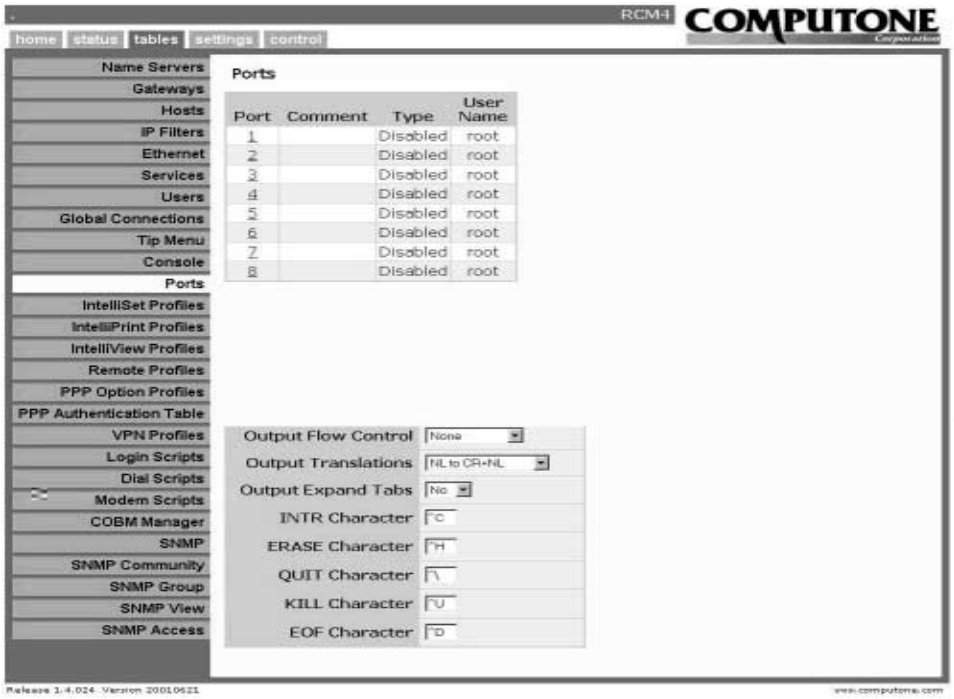


Figure 17 Ports Screen

Configure serial port parameters by using the ***Ports*** configuration screen. When changing the configuration of a port, the changes take effect the *next* time the port is opened. For example, if a user is currently logged into a port and the line speed is changed, the change does not take effect until that user logs off. The new login prompt is issued at the new line speed. An exception is when changing the configuration of the port that is running. Then an option is given to allow the changes to take place immediately (but return to original values at next login), to take effect at next login, or both.

Port Type—How Will the Port be Used?

Using this screen, configure the RCM 4/8 on how each port will be used. Is someone to be logged into this port? Is someone on the network to be able to dial

out on a modem attached to this port? Is a printer to be attached? Want to start up a PPP/SLIP link? There are four port types that support terminals and dial-in connections:

- Login by Port
- Login by Screen
- Auto-Login
- Auto-Login/Wait

For these types, the connection is started by whatever is attached to the serial port. This may be a person sitting at a local terminal, a client who dials into an attached modem, or a computer that dials in and sets up a PPP connection by running a login script. When modems are used, the ports are usually configured so they detect incoming calls by waiting for the modem to assert *carrier* (**DCD**), and when *carrier* is dropped, to recognize that the connection has been dropped.

There are two port types which can support connections that are started by processes running elsewhere on the network, such as printing and dial-out capabilities:

- Reverse-TCP
- Printer

There is one type which supports dial-in and dial-out connections:

- Login by Port/TCP

There is one port type which supports dial-out PPP/SLIP/CSLIP connections to other networks:

- Out-bound Connection

There is one last port type which supports remote serial port access to the RCM 4/8 in a Windows® NT environment:

- Remote Serial Port

Configuring a Port

To configure a port, click on the port number shown in Figure 17 , and the ***Port Configuration*** screen appears:.

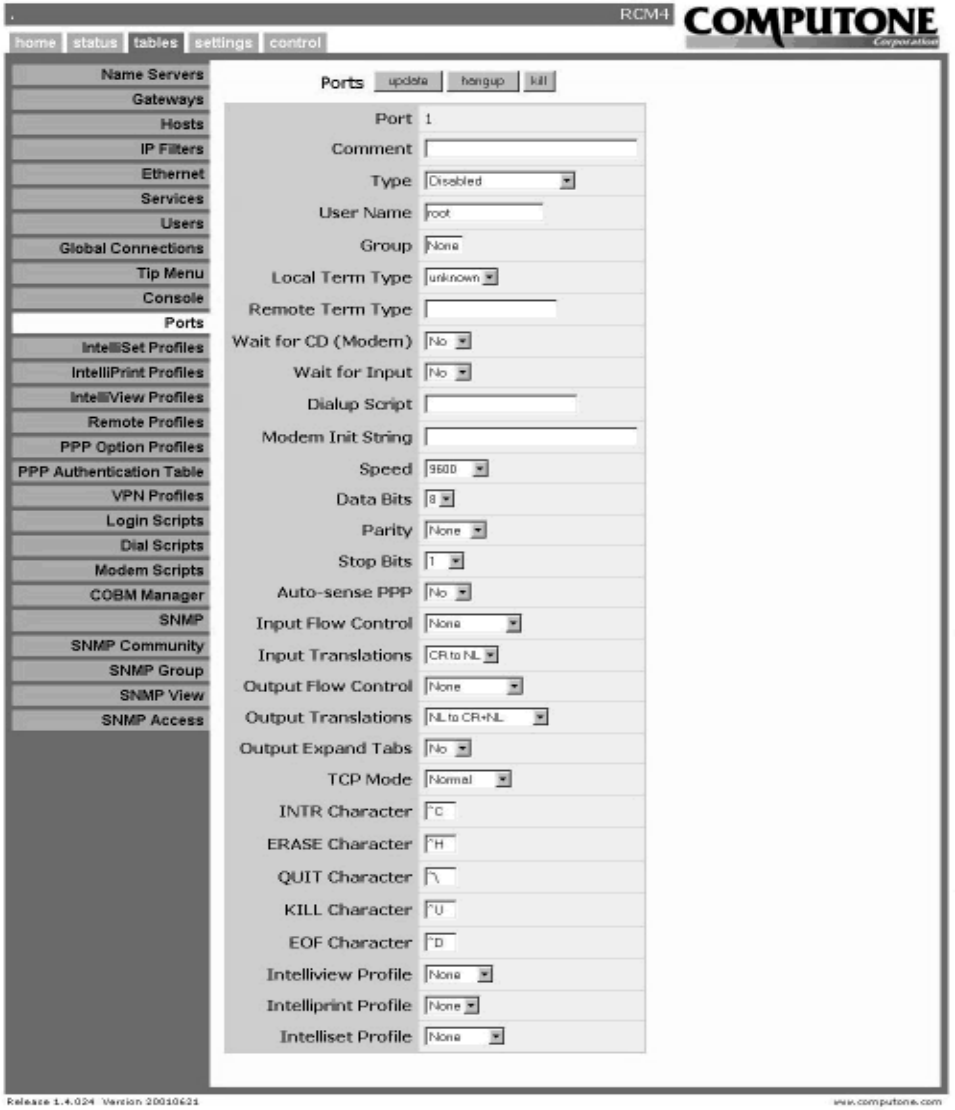


Figure 18 Port Configuration Screen

The following table defines the port configuration parameters:

Table 6 Port Configuration Parameters

Keyword	Parameters	Definition
Port		Number of port being configured
Comment		Some descriptive comment to identify the port.
Type		
	Disabled	Nothing happens on this port, except the ability to use commands such as **tip** and **output** to send data to it in order to test the port or configure a modem or other device.
	Login by Port, Wait	With this selection, the port sends a login prompt to the attached terminal or modem. When the user logs in, the RCM 4/8 starts up whatever connections have been configured for that user.
	Login by Virtual Screen	Generally use this setting only if the port is configured to support multiple sessions through IntelliView. Each virtual screen is sent its own login prompt, and the user must log into each virtual screen separately. When a session is ended, a new login prompt for that virtual screen is sent, but DTR is not dropped if there is any other session on this port still active.
	Auto-Login/Wait	This is almost identical to *Auto-Login*, except that instead of launching the connection immediately, the port first sends a prompt: **Press <enter> to continue**, and when the operator does this, *then* the connection is launched. This is designed to solve the quandary that occurs when a port configured as *Auto-Login* is attached to a local terminal that is always on but frequently unattended. The user logs off and walks away, and the RCM 4/8 immediately launches the connection. Suppose that connection is an attempt to rlogin to some host machine. So that machine prompts for a password. Since there is no one present to enter a password, the connection soon times out and is restarted, and times out and is restarted and so on. If the port is configured as *Auto-Login, wait*, then the RCM 4/8 remains at the "**Press <enter> to continue**" prompt until someone does this and the retries and time outs are avoided.
	Auto-Login	To configure a port so that the RCM 4/8 automatically starts a connection without prompting for a login, perform the following step. If the port is configured for *Auto-Login*, a user name for this port (*myname*, in the above example) must also be specified.The port would behave exactly as a *Login-by-Port* but instead of sending a login prompt, he assumes that the specified user has successfully logged in, and starts up his connections accordingly. When the session is over, the RCM 4/8 will (after waiting for *carrier* when appropriate) restart the sessions again.
	Printer	This configuration is similar to *Reverse-TCP*, except that a port configured as a *printer* can also accept connections from **rcp** and **rsh cat** clients on the network.
	Reverse-TCP	When a port is configured as *Reverse-TCP*, the RCM 4/8 accepts a TCP connection from some other host on the network. Data received from that host is sent out the serial port, and data received from the serial port is sent to the host. This is a common method of supporting printers and other "non-login" serial devices.

Table 6 Port Configuration Parameters

Keyword	Parameters	Definition
	Out-bound PPP or SLIP Connection	This configuration supports outbound PPP/SLIP/CSLIP links. The RCM 4/8 brings these links up automatically when it tries to route a packet to a network that it knows to be on the other side of one of these links. Note that this port type supports only *dial-out* connections. To support clients who are dialing into the RCM 4/8, configure the port as *Login-by-Port*.
	Login by Port/TCP	This is a combination of *Login-by-Port* and *Reverse-TCP*, and is designed to support bidirectional operation of a modem. Configure the port as *modem enabled*, because the RCM 4/8 uses *carrier* (DCD) to sense incoming calls and determine whether there has been a disconnection. When the port is idle and there is no incoming call, the RCM 4/8 accepts TCP connections for this port from hosts on the net, just like *Reverse-TCP*. If a connection is established, the client can access the modem, send dialing commands, and connect to other systems. Anyone trying to dial in gets a busy signal because the modem is off-hook. If an incoming call comes in first, the port sends out a login prompt, like *Login-by-Port*, and as long as the incoming call is connected, the RCM 4/8 refuses or defers TCP connections from the network for that port.
	Remote Serial Port	When a port is configured as a remote serial port, it can establish communications with another host's driver. Once a host's driver establishes communication with a RCM 4/8 PowerRack, it attaches to one or more remote serial ports (RSP) and presents them as "logical" serial ports to the host operating system. In the case of NT, these appear as COM ports.
User Name		Name of port user.
Group		There are 16 groups of ports, numbered 0 to 15. Any port can belong to any group, or to no group at all. When something tries to start a reverse-TCP connection to the RCM 4/8, it can specify a particular port or a particular port group. When a port group is specified, the first available port in the group is used. A port group number can also be specified in a *Remote Profile* for an outbound PPP/SLIP/CSLIP interface. A port is configured as *Reverse-TCP* or *Login-by-Port/TCP* cannot be a member of the same group as a port configured as *Printer* or which uses *IntelliPrint*. This is because the first types suppress output processing, while the others perform it. If both types were members of the same group, the results might depend on which printer happened to be available.
Local Term Type	Selections are: unknown , wyse30 , xterm , ansi , wyse50, uterm0, uterm1, vt100, wyse60, uterm2, uterm3	This setting defines the terminal characteristics that will be used when the RCM 4/8's menu interface is running on this port. This also defines the default terminal name that is sent when telneted or rlogined from this port to a host on the network. This default value may be overridden by other settings, however. Because this information is used by the menus, the RCM 4/8 needs to understand the terminal characteristics that each terminal name represents. For that reason, there are a limited number of these supported. The last four terminal types are user-definable. If the terminal does not emulate one of the defined terminals, the terminal's definitions can be stored under one of these four terminal types.

Table 6 Port Configuration Parameters

Keyword	Parameters	Definition
Remote Term Type		Enter a name here, then by default it IS sent when rlogined or telneted to a host, instead of using the one given for the *Local terminal name*. Since the RCM 4/8 does not need to know what this name actually means, it can be any name that is understood by the login host. The telnet and rlogin commands also support command-line arguments which, if used, can override these default terminal-types. If there is no command-line argument, the *remote term type* is used, and if no *remote term type* is defined, then the *local term type* is sent.
Wait for CD (Modem)	Yes/No	The modem port waits for the modem to assert carrier (**DCD**), if the port was configured as a *modem port*. When a *modem port* is connected to a local terminal, the port's **DCD** is usually wired to the terminal's **DTR** or **RTS** (whichever is *not* being used for flow-control). In that case, the RCM 4/8 would be waiting not for an incoming call, but for someone to turn the terminal on. If this is a non-modem port, it is assumed that carrier is present. The RCM 4/8 waits for 1 second before continuing, after carrier (**DCD**) is detected by a *modem port*.
Wait for Input	Yes/No	This allows any attached modem or device to stabilize, before an attempt to send data to it. For example, there are some modems which assert carrier before coming out of command mode. Data intended for transmission to the remote modem may be interpreted as a command. After this 1 second delay, any data that might have been received so far is flushed before the preamble or login prompt is sent.
Dialup Script		This is used by ports configured for outbound PPP/SLIP/CSLIP links. It specifies the commands that have to be sent to the modem so it dials and establishes a connection and allows the RCM 4/8 to wait for particular responses. Different modems may require different dialer scripts; that is why the dialer script is stored on a per-port basis, while the *login script* (which depends on the particular target of the call) is identified in the *remote profile*.
Modem Init String		This setting is used by ports that are configured for terminals or dial-in connections. It defines a string of commands which the RCM 4/8 transmits to the modem before it waits for the next incoming call. This is not always required. Some modems can be configured ahead of time and never seem to lose their settings. When does the string get sent: some user logs off, RCM 4/8 drops **DTR** to hang up the line, waits a second, raises **DTR**, *sends the initialization string*, waits for modem to assert **DCD**, call comes in, next fellow logs in, works, logs off—and it starts all over again.
Speed	Set speed	This sets the line speed at which data is transmitted and received. In addition, custom rates can be defined by setting up an IntelliSet profile and assigning that profile to a port. By using IntelliSet, a *split baud-rate* can also be specified where the port transmits at one speed, and receives at another. When line speeds and other parameters are defined using IntelliSet, those values override the ones chosen here. 50 150 1200 3600 19200 64000 230.4k 75 200 1800 4800 38400 76800 307.2k 110 300 2000 7200 56000 115200 460.8k 134.5 600 2400 9600 57600 153.6k 921.6k

Table 6 Port Configuration Parameters

Keyword	Parameters	Definition
Data Bits	Selections are 8, 7, 6, and 5.	How many data bits per character.
Parity	Selections are: odd, even, space, mark, and none.	This controls the parity bit sent with each characters.
Stop Bits	Selections are: 1, 1.5, or 2 bits.	This controls the number of stop bits that are transmitted after each character. One stop bit is generally sufficient except when connecting to devices that are very old, very slow, or very unusual. This has no effect on the receiver. One stop bit is always sufficient.
Auto-sense PPP	Yes/No	This controls whether this port senses a PPP packet and automatically starts up a PPP link.
Input Flow Control	Selections are: none, XOFF, RTS, and XOFF & RTS.	Since the RCM 4/8 can receive data from a serial device, it must be configured to signal when its buffers start to fill up so that the serial device stops sending data for a while.
Input Translations	Selections are: CR to NL (carriage return to newline), none (no translations), and NL to CR (newline to carriage return).	Any input processing specified here affects the port's operation when it is accepting line-based input, such as at the command prompt, or when in telnet command mode. At other times, the individual applications (telnet, rlogin, menu, etc.) force the input processing to an appropriate setting.
Output Flow Control	Selections are XON, None, XANY, CTS, XON & CTS, and XANY & CTS.	Sometimes the serial port on the RCM 4/8 is the sender, and it must avoid overrunning the terminal, modem, or printer to which it is attached. This is called *output flow control*. The *output flow control* chosen needs to match the *input flow control* of the device that data is being sent to, and vice-versa.
Output Translations	Selections are NL to CR+NL, None, CR to NL, Strip CR, CR | NL to CR+NL, and NL to CR+NL.	Any output processing specified here affects the operation of the port only when it is configured as a printer.
Output Expand Tabs	Yes/No	With this setting, the port will translate ascii **tab** characters to a sequence of spaces sufficient to achieve tab stops at 8-character intervals. This tab setting corresponds to the traditional tab processing performed on UNIX systems and is useful when printing output from a UNIX system using tools that expect this processing to be performed "downstream". If this parameter is set to *No* (or *disabled* using the command), then **tab** characters are sent unchanged.

Table 6 Port Configuration Parameters

Keyword	Parameters	Definition
TCP Mode	Selections are: Normal, CRNL ->CR, and Raw.	A protocol which establishes a reliable connection between two processes, generally on separate computers. Higher-level protocols like telnet and rlogin rely on TCP to ensure that data is not lost in transmission and that data is not sent faster than it can be processed. Normally, a reverse-TCP connection uses telnet protocol. Telnet server implementations differ in their treatment of carriage-return (CR) and linefeed (or new-line, NL) characters. With some, if a CR-NL pair is received from the network, the two characters will be output. That is what the *normal* option does. With other telnet servers, if a CR-NL pair is received, the CR is sent but the NL is ignored. This ia the *CRNL->CR* option. These two options are provided for maximum compatibility. The third option, *Raw*, causes the Reverse-TCP connection on that port to not use telnet protocol at all. Instead, the data received over the TCP connection is sent to the port exactly as received, and vice-versa. This is provided for compatibility with other vendors' products, as well as providing an easy-to-use interface for special applications.
INTR Character	^C	This defines the *interrupt key*. Use this key to quickly terminate commands before they have finished. In this example **^c** represents *control-c*.
ERASE Character	^H	This defines the character used to backspace a single character and erase it.
QUIT Character	^\	This defines the quit character.
KILL Character	^U	This defines the kill character.
EOF Character	^D	This defines the character used to denote the end of file character.
IntelliView Profile	None/Already created	This specifies the name of the IntelliView profile that is wanted to apply to this port. It can be any profile that has been already created.
IntelliPrint Profile	None/Already created	This specifies the name of the IntelliPrint profile that is wanted to apply to this port. It can be any profile that has been already created.
IntelliSet Profile	None/Already created	This specifies the name of the IntelliSet profile that is wanted to apply to this port.

IntelliSet Profiles

Selecting the ***IntelliSet Profiles*** parameter displays the following screen:

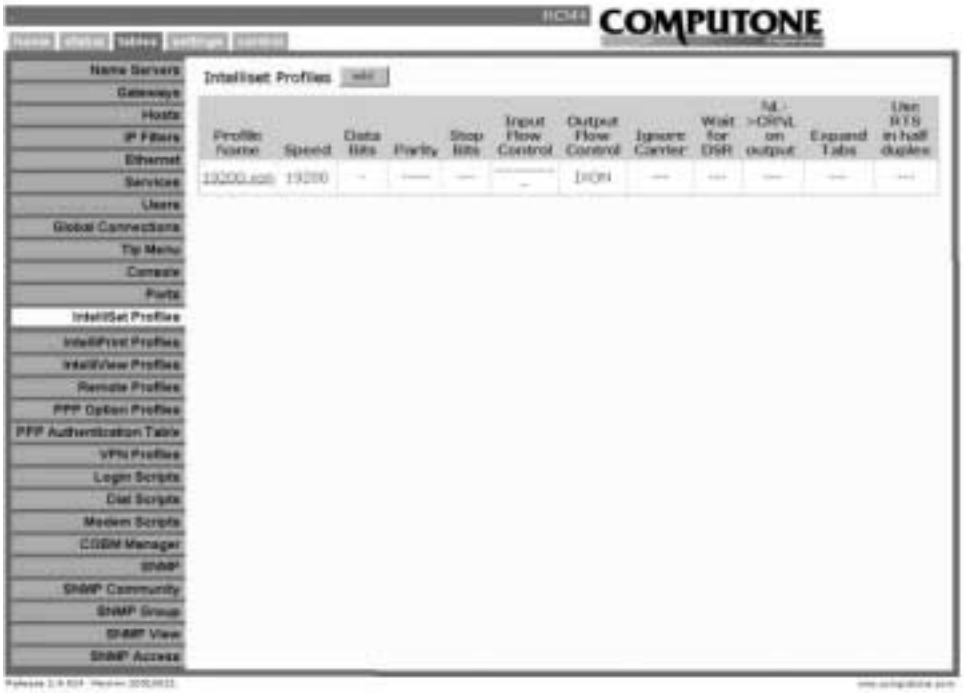


Figure 19 IntelliSet Profiles Screen

IntelliSet profiles include a mixed bag of specifications that can be thought of as an extension of port configuration. In fact, there are many parameters in common. The difference is that features specified with IntelliSet *override* the settings in port configuration.

IntelliSet parameters also resist any attempts by applications to change them. For example, a telnet session normally puts a serial port into raw mode, disabling any output processing that might have been specified under port configuration. If, however, this had been specified in an IntelliSet profile, the output processing

continues. To change the ***IntelliSet Profile*** parameters, click on the profile name, and the ***IntelliSet Configuration*** screen appears:.

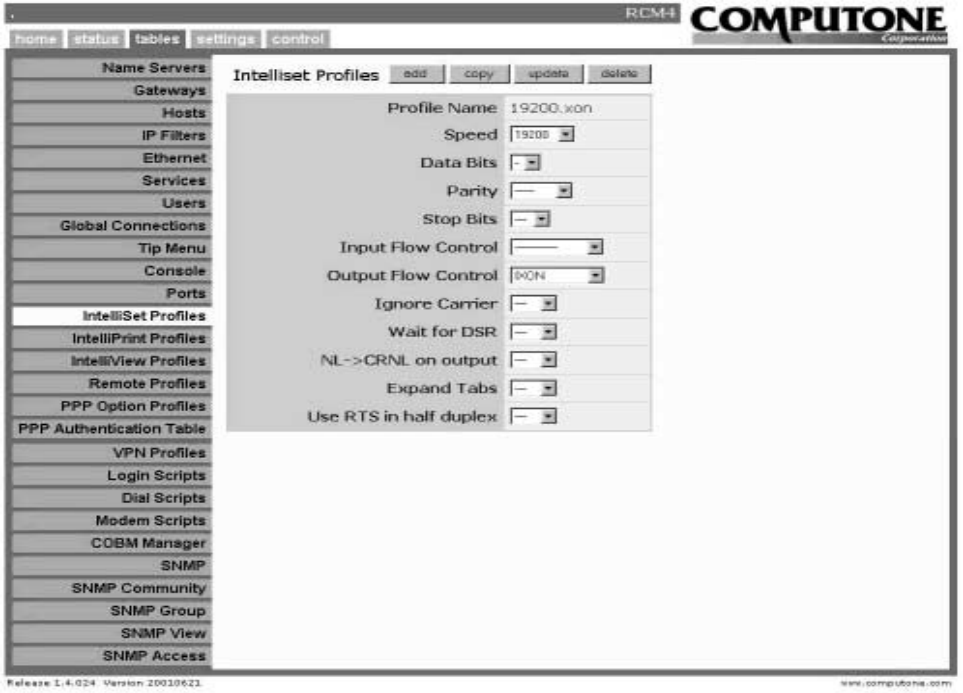


Figure 20 IntelliSet Configuration Screen

The IntelliSet Configuration Form is shown in Figure 20 , and illustrates profile *19200.xon* which is supplied with the factory defaults. Except for the profile name and the two custom baud rates, all the other fields are pick-lists and offer many of the same selections as the Port Configuration screen. In this example, many of the input areas contain dashes. These indicate that the corresponding parameter is not specified by this IntelliSet profile. When a value appears instead of dashes, then that parameter is specified. When a parameter is specified in an IntelliSet profile, that value overrides anything in the port configuration and it cannot be changed by whatever application might be running (telnet, rlogin, etc.).

Table 7 IntelliSet Parameter Selections

Keywords	Parameter	Description
Profile Name		The name of the IntelliSet Profile under configuration.
Speed	Set baud rates.	The following baud rates are available: ------ 50 150 1200 3600 19200 64000 75 200 1800 4800 38400 76800 110 300 2000 7200 56000 115200 134.5 600 2400 9600 57600
Data Bits	Selections are: ---, 5, 6, 7,and 8.	The data bits selections that are available.
Parity	Selections are: odd, even, space, mark, and none.	This controls the parity bit sent with each characters.
Stop Bits	Selections are: --, 1, 1.5, 2.	The stop bit selections that are available.
Input Flow Control	Selections are: -----, None, IXOFF, RTS, RTS+IXOFF.	*IXOFF* indicates that the RCM 4/8 should send an XOFF character when its receive buffers become nearly full, and send an XON character when they again have room for more data. *RTS* indicates that the RCM 4/8 should drop the RTS signal when the buffers are nearly full, and raise it when they have room. RTS+IXOFF indicate that a combination of actions are taken.
Output Flow Control	Selections are: -----, None, IXON, XANY, CTS, IXON+CTS, IXANY+CTS.	The *Outflow* parameter allows the output flow control to be specified and locked. When the RCM 4/8 is sending data to a device, and that device cannot process data quickly enough, it must signal the RCM 4/8 in some way to tell it to stop transmitting. Output flow control specifies what conditions cause the RCM 4/8 to stop sending data. The dash indicates that this IntelliSet profile will not affect output flow control. *CTS* indicates that the RCM 4/8 will not transmit unless the CTS input is asserted. *IXON* indicates that the RCM 4/8 should stop transmitting when it receives an XOFF character and resume when it receives an XON. *IXANY* is the same, except that after the XOFF character has disabled transmission, receiving *any* character (not just an XON) will re-start it.
Ignore Carrier	Selections are: -----, Yes, and No.	The *Ignore Carrier* parameter specifies whether the port will be treated as a *modem* port. Modem ports are affected by the DCD (carrier detect) signal, while non-modem ports ignore it. The dash indicates that this IntelliSet profile will have no effect on whether the port is a modem or non-modem port.

Table 7 IntelliSet Parameter Selections

Keywords	Parameter	Description
Wait for DSR	Yes/No	The *data-terminal-ready* (**DTR**) signal from the RCM 4/8 is asserted when the port is opened, dropped when the port is closed, and stays dropped if the port is disabled or the RCM 4/8 is off. To prevent the modems from answering the phones when the RCM 4/8 Communications Server is turned off, be sure **DTR** is connected and make sure that the modem has not been configured to ignore **DTR**. If using CTS/RTS flow control be sure to connect both of these signals as well. The RCM 4/8 software does not require the use of the *data-set-ready* (**DSR**) and *ring-indicator* (**RI**) signals, so these do not need to be connected.
NL->CRNL on Output	Yes/No	These parameters allow the output processing on this port to be specified and locked. Carriage returns can be inserted before linefeeds to prevent *barber-pole* output. The dash indicates that this IntelliSet profile should not affect output processing.
Expand Tabs	Yes/No	Tab expansion can be helpful if the output contains tab characters and the terminal doesn't understand tabs.
Use RTS in Half Duplex	Yes/No	*RTS* indicates that the RCM 4/8 should drop the RTS signal when the buffers are nearly full, and raise it when they have room.

IntelliPrint Profiles

Selecting th***e IntelliPrint Profiles*** parameter displays the following screen:

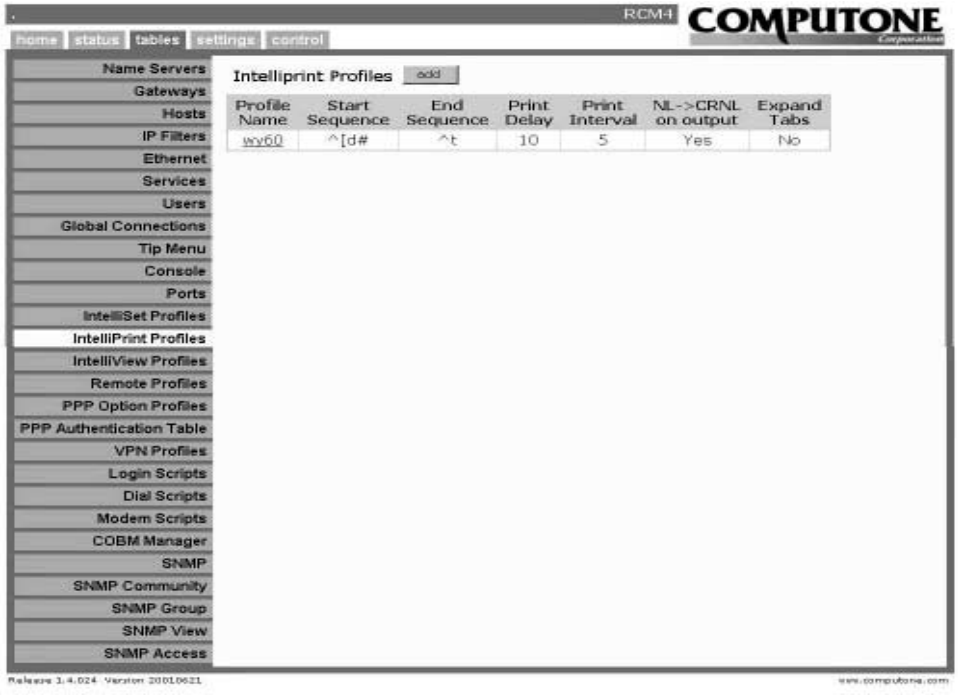


Figure 21 IntelliPrint Profiles Screen

To support IntelliPrint, the terminal must have an AUX port, and support commands that route subsequent data to that port instead of to the display, and also route subsequent data to the display and no longer to the Aux port. These commands are special data sequences that are sent from the RCM 4/8 before and after any data directed to the printer. An IntelliPrint profile contains these sequences, as well as other information for setting the relative priorities between data for the printer and data for display.

When an IntelliPrint profile with a serial port has been associated an IntelliPrint, configure the network hosts to send data directly to that printer, independently of what is happening on the terminal. For example, the terminal could be logged into a host and running an application. While in the application, a decision is made to print a report. The report is sent to the system's print spooler, which has

been configured to send output to the printer attached to the terminal. While the output is printing, the terminal is still able to be used.

To configure the IntelliPrint profile, click on the profile name and the ***IntelliPrint Profiles Configuration*** screen appears:

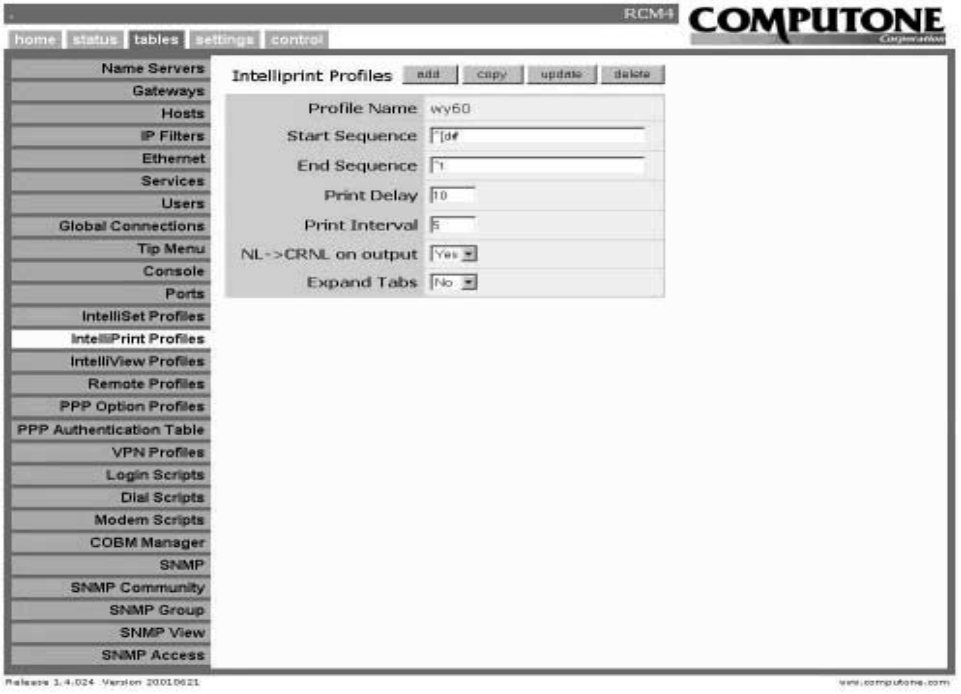


Figure 22 IntelliPrint Profiles Configuration Screen

The IntelliPrint Profile parameters are defined as follows:

Table 8 IntelliSet Parameter Selections

Keywords	Parameter	Description
Profile Name		The name of the IntelliSet Profile under configuration.
Start Sequence	Type a command.	To make a terminal send data to its Auxiliary port, the RCM 4/8 must first send a command saying "*send all subsequent data to the aux port, until I say otherwise*". Not in so many words, of course, but some sort of special data sequence must be used. In this example, for the Wyse 60 the sequence consists of the three characters: **escape d #**. The escape character is represented here by the symbols **^[**. It is often necessary to represent the *escape* code and other unprintable bytes in IntelliFeatures profiles. How are the codes to use for a particular terminal determined? Consult the terminal's manual or programmers' guide. What is called the *start print sequence* is often called *start transparent print*. The print sequences to use are determined by which *terminal* is being used, not by the choice of printer. The printer never "sees" these sequences, they are interpreted by the terminal.
End Sequence	Type a command.	When the RCM 4/8 has been sending data to the printer and now wants to send data to be displayed again, it must send a command to the terminal. For a Wyse 60, the command to do this is the single character, **ctrl-t.** If data is to be sent to a printer attached to the terminal, that data had better *not* contain the terminal's *end print* sequence. When the terminal sees such data, it will *not* send it blindly to the printer. It will rightly interpret it as a command and send the following data (that had been intended for the printer) to the display.
Print Delay	Type a command.	The *Print Delay* tells how long the RCM 4/8 must wait after any display output before it sends any data for the printer and the delay is measured in tenths of a second. Why a delay? Data for display often contains control sequences for cursor addressing, highlighting, and so on. These sequences consist of two or more bytes of data and most terminals get confused if such a sequence is interrupted by a command to start transparent printing. The delay ensures that all the bytes of any control sequence have a chance to be completely sent before a command to start printing is sent.
Print Interval	Type a command.	The *Print Interval* defines a delay to be inserted between successive blocks of print data and is measured in tenth-seconds. Why a delay between successive blocks of print data? To make the RCM 4/8 sends its printer data more slowly. Most terminals can display faster than most printers can print.

Table 8 IntelliSet Parameter Selections

Keywords	Parameter	Description
NL->CRNL on Output	Yes/No	This defines whether the RCM 4/8 adds Carriage-Returns before linefeeds and expand tabs in data to be sent to the printer. In most cases it is better to keep this option disabled and configure the host software to perform whatever processing is appropriate.
Expand Tabs	Yes/No	This defines whether the RCM 4/8 adds expand tabs in data to be sent to the printer. In most cases it is better to keep this option disabled and configure the host software to perform whatever processing is appropriate.

IntelliView Profiles

Selecting the ***IntelliView Profiles*** parameter displays the following screen:

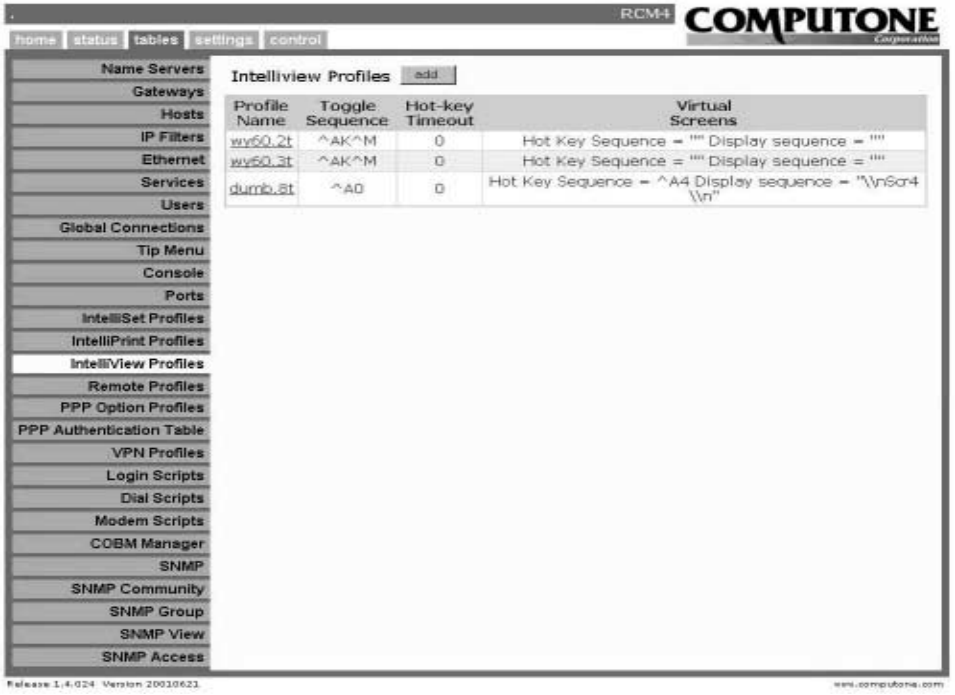


Figure 23 IntelliView Profiles Screen

IntelliView allows a running of a separate session on each of the terminal's display pages. On one page (or "screen") a host might be logged in, while on another page a different host is logged in. Use designated function keys to flip between the two sessions. In order to use the terminal's function keys to flip between pages, the RCM 4/8 needs to be told what codes these keys send. Generally this is not a problem, but there are some applications which send commands to the terminal to reprogram its function keys to specific values. If this includes a function key desired to use for IntelliView, it may inhibit the switching between screens once an application is entered. In that case, reconfigure that application or use a different function key for IntelliView.

To support IntelliView, the terminal must be capable of maintaining two or more *pages* of display memory. Often, the number of pages supported depends on the display mode. For example, a Wyse 60 in 132x43 mode (Wy60 emulation) might

support only a single *page*, where in 80x25 (Wy60 emulation) it might support two pages, and in “Econ-80” mode it might support three or more pages. If not sure about the particular terminal, read its manual.

The WY60.2t IntelliView profile, shown in Figure 23 , supports a Wyse 60 terminal with two pages of screen memory. Three *hot keys* are defined, all of them function keys:

- Press **F12** to switch between screens. When pressed, the three-byte sequence **^A K ^M** is sent.
- Press **Ctrl-F1** to switch to screen 0 (the main screen). If the Wy60 is configured for 8-bit data, this key sends a single byte, octal value 200.
- Press **Ctrl-F2** to switch to screen 1. This key sends a single byte of octal value 201. If creating a new IntelliView profile, enter the name on the first line. If modifying an existing one, the name is already there and changes are not allowed.

To edit one of the profiles, click on the profile name (wy60.2t is selected, for this example), and the ***Hot Key Sequence*** screen appears:

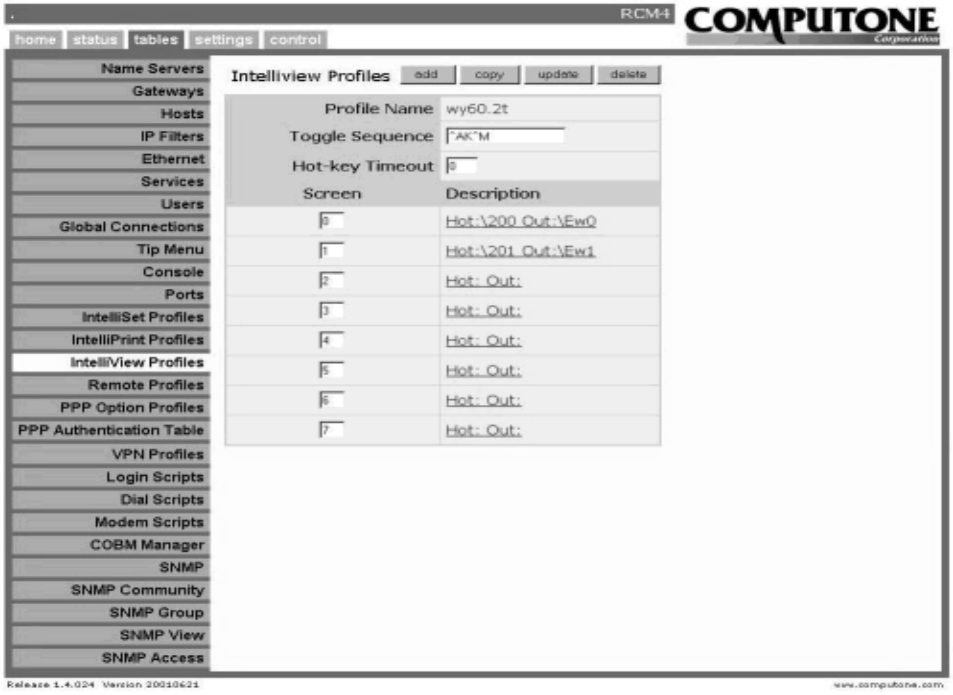


Figure 24 Hot Key Sequence Screen

This screen allows the entry or change of the *Hotkey Sequence* and *Select Sequence* octal values.

Remote Profiles

Selecting the ***Remote Profiles*** parameter displays the following screen:

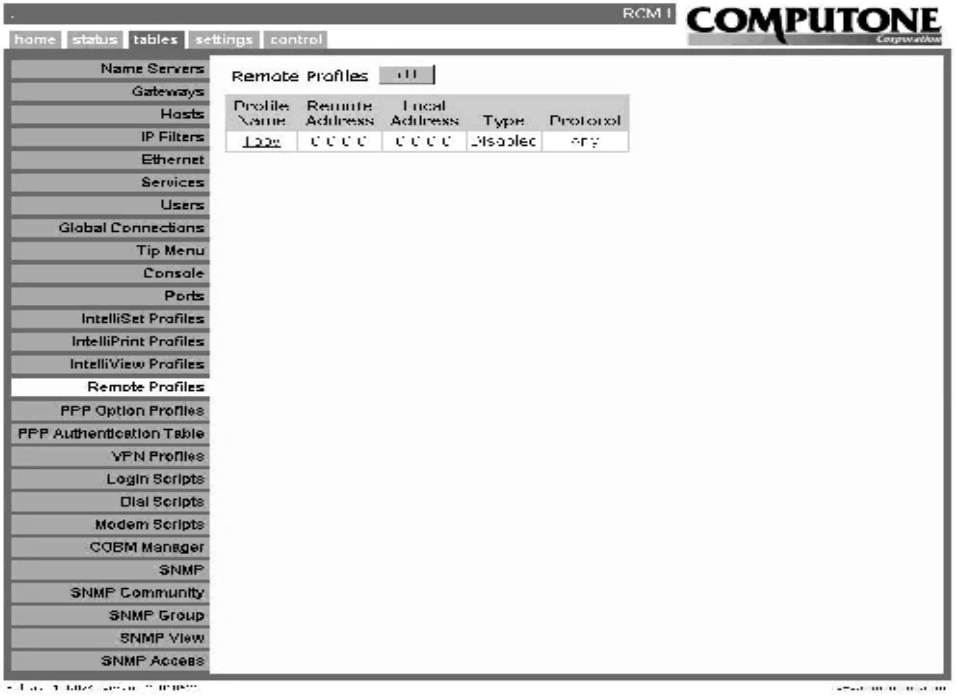


Figure 25 Remote Profiles Screen

Remote Profiles contain basic information about the interface: whether it is inbound (i.e., dial-up) or outbound, whether it is dedicated to a particular serial port or user, which protocols may be used, the IP address of the remote site, and so on. To add a profile, click ***add*** then enter a name for the remote profile. Once the profile is created, click on the name to change the values (in this instance, ***Toby*** is used as an example), and the ***Remote Profiles Parameter*** screen appears:

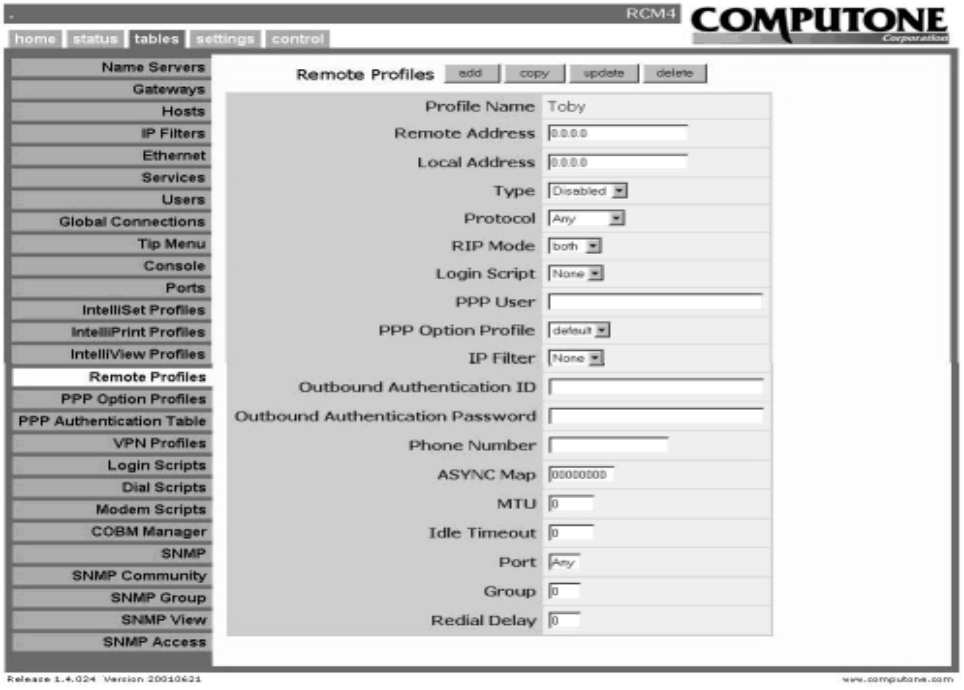


Figure 26 Remote Profiles Parameters Screen

Table 9 Explains parameter selections.

Table 9 Remote Profile Parameter Selections

Keywords	Parameter	Description
Profile Name		The name of the PPP Option Profile under configuration.
Remote Address	Type address	The *Remote Address* is the IP address of the PPP, SLIP, or CSLIP interface at the other end of the link. When the link is brought up, this address is used unless a different one has been assigned through PPP address negotiation or information from the RADIUS user file. It is possible to leave this field set to **0.0.0.0**, in which case the correct IP address *must* be supplied by other means. Here are the rules: • For *Outbound* interfaces, the *Remote Address* must be set to the correct value, because it is the attempt to route to this address that brings up the link. This address cannot be subsequently changed by PPP address negotiation. • For *Inbound* interfaces, if the IP address of the remote interface is supplied from the RADIUS user database, or if it will be available from PPP address negotiation, the *Remote Address* in the Remote Profile can be left "open", i.e., set to **0.0.0.0**. • For *Inbound* interfaces, if the IP address of the remote interface will not be available by other means, the *Remote Address* must be set to some valid address. This technique is widely used by ISP providers to supply temporary IP addresses to dial-in users. The *Remote Addresses* that has been assigned to various inbound interfaces comprise a pool of available IP addresses that are assigned dynamically as users dial in.
Local Address	Type address	The *Local Address* is the IP address of this end of the PPP, SLIP, or CSLIP link. If two RCM 4/8s were connected via a PPP connection, each one's *Local Address* would be the other's *Remote Address*. The *Local Address* must be set to some valid address, but ***Local Addresses* in different Remote Profiles are not required to be different.** In fact, it is common for Internet Providers to use the Ethernet's IP address for all interfaces. In some situations, a different IP address may be needed. For instance, this could be an outbound interface to a site which expects to have a particular IP address. This could happen if the remote site had another RCM 4/8 and it were configured to assign a specific IP address for specific users.
Type	Selections are ***Disabled***, ***Inbound, Outbound***, or ***Both.***	The *Interface Type* specifies whether this Remote Profile's interface will support inbound connections, or initiate outbound connections. The default value is **disabled**, so it must be set before the interface can be used.

Table 9 Remote Profile Parameter Selections

Keywords	Parameter	Description
Protocol	Selections are Disabled, PPP, SLIP, CSLIP, and Any.	For an ***Outbound*** interface, either SLIP, CSLIP, or PPP must be specified, because the interface needs to know which protocol to use before it can bring up the link. For an ***Inbound*** interface, the desired protocol is learned directly from the user. If it is an NVRAM user it is configured specifically as either a SLIP, PPP, or CSLIP user. If it is a RADIUS user, similar information is stored in that database. For inbound interfaces, then, this is used to optionally restrict this Remote Profile's use, much like the *Serial Port* and *Dial-in User* are. If the Remote Profile is to be available for any protocol, set the *Protocol* to **Any**. To restrict it, specify SLIP, CSLIP, or PPP. The setting **Disabled** exists for compatibility with earlier versions of the RCM 4/8. It is no longer needed because a Remote Profile's interface can be disabled by setting the *Interface Type* to **disabled.**
RIP Mode	Selections are: Any	RIP (Routing Information Protocol Mode) is used when the RCM 4/8 needs to share routing information with other hosts. By *listening*, it learns routes from other hosts and by broadcasting or *sending*, it tells other hosts about the routes *it* knows.
Login Script	Selections are earlier defined names.	The *Login Script* is used only by ***Outbound*** interfaces. It allows the interface to log into the remote host. An earlier defined name of one of the Login Scripts must be supplied. If this is left blank, then no login script is used. It is unusual for a dial-in connection to *not* require a login, so one must be specified.
Outbound Authentication ID		This setting applies to ***Outbound*** interfaces. For *Outbound* interfaces, the RCM 4/8 supplies the information when requested by the site that is logged into.
Outbound Authentication Password		This needs to be a matter of some pre-arrangement. If the RCM 4/8 has the outbound interface and the remote site expects CHAP authentication, it had better enable it and know the correct CHAP name and secret to configure. So, this is not always a question of how to configure things.
PPP User		When the user logs in, the intent may be to bring up a PPP or SLIP connection between the RCM 4/8's local network and a client computer that has just dialed in. These users are called ***PPP users*** (although they may be using SLIP or CSLIP protocol instead). Sometimes these are called ***framed users*** because PPP, SLIP, and CSLIP are all protocols in which data is *framed* (i.e., separated into well-defined blocks marked by headers). When configuring a ***PPP user***, provide networking information particular to this user so that routes between its network and yours will be set up correctly. A *framed user* exists only for the purpose of bringing up the PPP or SLIP link. Once the network has been extended by this connection, hosts on one side of this connection can connect to hosts on the other side. These connections may include rlogin and telnet sessions, and those users have no relationship to the *framed user* that caused the PPP/SLIP connection to be made.

Table 9 Remote Profile Parameter Selections

Keywords	Parameter	Description
PPP OPtion Profile		This has been defined in PPP Profile.
IP Filter	Type an earlier defined filter name.	This is the name of an IP filter that has been defined. If this is blank, no IP filter is attached and the traffic through this interface is unrestricted. It is common for different interfaces to have different filters. Certain things might be allowed in the local Ethernet Network might not be allowed through any of the remote interfaces. Other traffic might be allowed over an outbound interface to a remote branch of the same business, that would not be allowed over the interface that goes to an Internet service provider (ISP).
Phone Number		The *Phone Number* is used only by ***Outbound*** interfaces with dial or login scripts. If the script contains the command **%p**, this phone number is inserted at that point. Being able to configure the phone number here conserves dial scripts.
ASYNC Map		This field is predefined.
MTU	set between 0- 2000+	Defines the transmitted unit size.
Idle Timeout		The amount of idle time set before the connection is terminated.
Port		Shows the port number of the remote interface.
Group		Displays the associated group of the user.
Redial Delay		Time set before trying to redial and reconnect.

PPP Option Profiles

Selecting the ***PPP Option Profiles*** parameters displays the following screen:

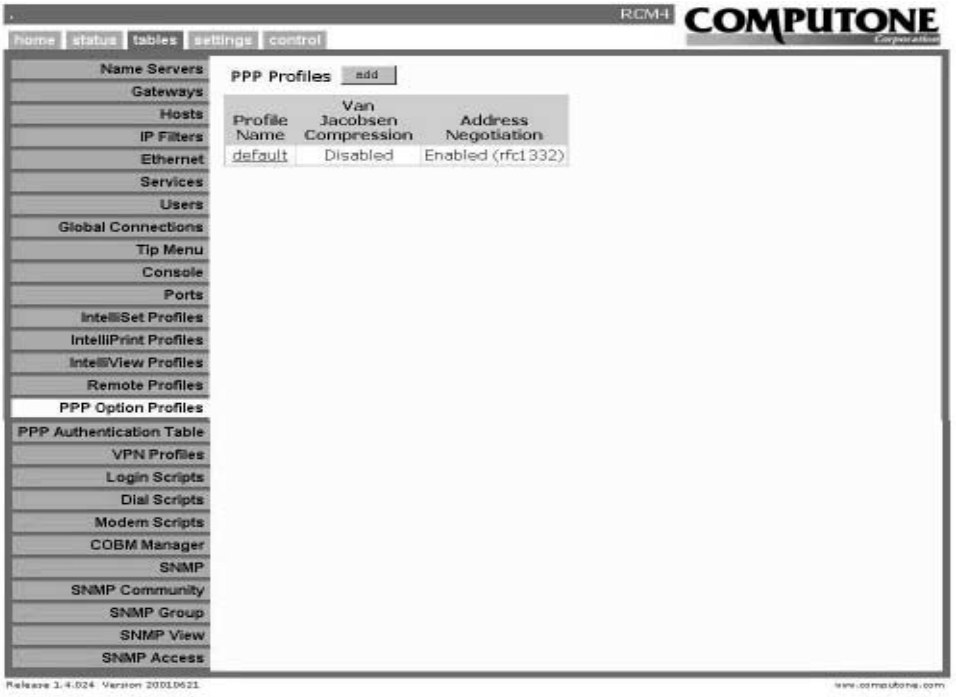


Figure 27 PPP Option Profiles Screen

PPP Option Profiles contain additional protocol options used in bringing up PPP and SLIP links. Options Profiles (sometimes called SLIP/PPP options) are used for storing configuration parameters that do not change often. These parameters are also likely to be shared by a number of interfaces at a given site.

An Options Profile is created with a particular collection of settings and it is given a name. To assign these settings to a particular interface, enter the name in that interface's Remote Profile. This reduces the number of separate parameters that an individual Remote Profile must contain. At factory default there is a single options profile defined called ***default***. To create a new profile, click on ***add.*** When the new remote profile is created (***test1***, in this example), this default profile is assigned to it and remains until it is changed.

To change PPP Option Profile ***test1***, click on it, and the ***Changing a Profile*** screen appears:.

Figure 28 Changing a Profile Screen

Table 13 describes the parameter selections.

Table 10 PPP Option Profile Parameter Selections

Keywords		Description
Profile Name		The name of the PPP Option Profile under configuration.
Van Jacobson Compression	Disable/Enable	*Van Jacobsen (VJ) Compression* is a method of compressing TCP/IP headers in PPP or SLIP packets. With SLIP protocol, both sides must agree beforehand whether to use it. (SLIP with VJ Compression is called CSLIP). With PPP, the two sides can negotiate whether to use VJ Compression. On the RCM 4/8, connections are designated PPP, SLIP, or CSLIP. Since CSLIP always uses VJ Compression, and SLIP never does, this option affects only PPP links. Set it to **Disabled** (the default) if VJ Compression is not used or to **Enabled** if it is used.

Table 10 PPP Option Profile Parameter Selections

Keywords		Description
Address Negotiation	Disable/Enable	Enabling *Address Negotiation* on inbound PPP connections allows the RCM 4/8 to learn the caller's IP address and inform the caller of our IP address through the PPP negotiation process. While address negotiation can be enabled for outbound connections as well, the RCM 4/8 needs to know the remote site's correct IP address ahead of time because it is the attempt to *access* that address which causes the RCM 4/8 to bring up the interface.
MRU Size	Type size	The *MRU Size* (*Maximum Receive Unit Size*) represents the maximum number of bytes the RCM 4/8 can receive in a single PPP packet. This is a partner to the MTU, or *Maximum Transmit Unit*, which is configured in the Remote Profile and defines the largest packet the RCM 4/8 can *send*. Each side of the link has an MRU, usually constrained by internal buffer sizes and an MTU. The first step is making sure that one side's MTU is not greater than the other side's MRU. With PPP, this is done through *Maximum Receive Negotiation.* If *Maximum Receive Negotiation* (or *mru*) is **Yes**, (and assuming the remote side of the link is so configured) each side informs the other of its own MRU. If the recipient's MTU is larger, it temporarily reduces it accordingly. For SLIP and CSLIP connections, the effective MRU is always 1536 bytes. A large value is chosen because there is no mechanism, other than mutual agreement at configuration time, to agree on a smaller value.
Prompt	Yes/No	This applies to inbound SLIP and CSLIP connections. When set to **Yes**, the RCM 4/8 prompts the user to enter his IP address. After the address is entered, the link is brought up using that IP address as the "Remote Address". This facilitates multiple sites being able to use a single interface at different times. This option does not apply to PPP connections which are able to use PPP address negotiation for this purpose. This option is also not required when remote dial-in users are configured on a RADIUS server because each user's IP address can be stored in the RADIUS server's user database.
Proxy	Yes/No	If the *Proxy* option is set to **Yes** (the default), the RCM 4/8 responds to ARP requests for the remote IP address on this interface, as long as the link is up. For example, suppose the RCM 4/8's IP address (on the local Ethernet network) was 160.77.99.30. Suppose the remote IP address (the host at the other end of this PPP link) was 160.77.99.17. If a host on the RCM 4/8's local network wanted to access this remote host it would think from the IP address that it is on the local network. So, it would perform an ARP request to learn the Ethernet Address. The RCM 4/8 replies giving its *own* Ethernet address and enabling it to receive packets destined for that host. If the option is set to **No**, Proxy ARP is not per-formed.
ACompress	Yes/No	The *Address Compression* controls the local compression of address and control fields in the PPP header. Specify **Yes** (the default) to compress these fields, or **No** to leave them uncompressed.

Table 10 PPP Option Profile Parameter Selections

Keywords		Description
PCompress	Yes/No	The *Protocol (Field) Compression* controls the local compression of the protocol field in the PPP header. Specify **Yes** (the default) to compress it, or **No** to leave it uncompressed.
Async	Yes/No	The *Async (Map)* is used by PPP to prevent certain control characters (such as XON and XOFF) from occurring in the data stream. The map indicates which characters are proscribed. Specify **Yes** (the default) to allow the RCM 4/8 to negotiate this map with the remote system. Specify **No** to force the RCM 4/8 to use the ASYNC Map specified in the Remote Profile.
Magic	Yes/No	The *Magic (number)* is a arbitrary 32-bit number which is randomly chosen by each side of a PPP link. During negotiation, each side sends the other its *magic number*. It would be unusual for two different hosts to randomly choose the same random number, so if the magic number received is the same as our own, it is assumed that something has gone wrong (perhaps the modem is running in loop-back mode) and the RCM 4/8 must be talking to itself. Since this is a bad thing, the RCM 4/8 drops the connection (hang up the modem, etc.). Choose **Yes** (the default) if the RCM 4/8 is to check the magic numbers, or **No** if it is to be ignored.
Passive	Yes/No	*Passive (Mode)* only affects an outbound PPP connection on the RCM 4/8. Specify **No** (the default) if it should initiate the PPP negotiations, or **Yes** if it should passively wait for the other side to do so.This option has no effect on SLIP or CSLIP connections because these protocols do not involve negotiations.
MRU	Yes/No	Each side of the link has an MRU, usually constrained by internal buffer sizes and an MTU. The first step to harmony is making sure that one side's MTU is not greater than the other side's MRU. With PPP, this is done through *Maximum Receive Negotiation.* If *Maximum Receive Negotiation* (or *mru*) is **Yes**, (and assuming the remote side of the link is so configured) each side informs the other of its own MRU. If the recipient's MTU is larger, it temporarily reduces it accordingly.
Bring Up	Yes/No	The *Bring Up (Slip Link Immediately)* option applies to outbound SLIP and CSLIP connections. By default this option is set to **No** and the RCM 4/8 attempts to bring up the outbound link when it is first required to route network traffic to the IP address at the other end. If **Yes** is selected, then the RCM 4/8 attempts to bring up the line immediately on start-up. Furthermore, if the link goes down (because of a modem disconnect, for example) the RCM 4/8 attempts to bring it up immediately.
Default Authentication	PAP, CHAP, MSCHAPv1, MSCHAPv2, Any	These choices are a security method to authenticate the user. Any allows no security, and PAP to MSCHAPv2 provides increasing security requirements.

PPP Authentication Table

Selecting the ***PPP Authentication Table*** parameter displays the following screen:

:

Figure 29 PPP Authentication Screen

PPP Authentication allows the viewing and changing of a user's security level access into the system. To change a user's security level, click on the ***Login Name*** and the following screen appears*:*

Figure 30 PPP Configuration Screen

The *Authentication Protocol* options are:

- ***Disabled***-This blocks out the user.
- ***PAP***-The lowest security level.
- ***CHAP***- The next level in security.
- ***MSCHAPv1***- Increased security level.
- ***MSCHAPv2***- The highest level of security.
- ***ANY***- No security, allows any user on the system.

VPN Profiles

Selecting the ***VPN Profiles*** parameter displays the following screen:

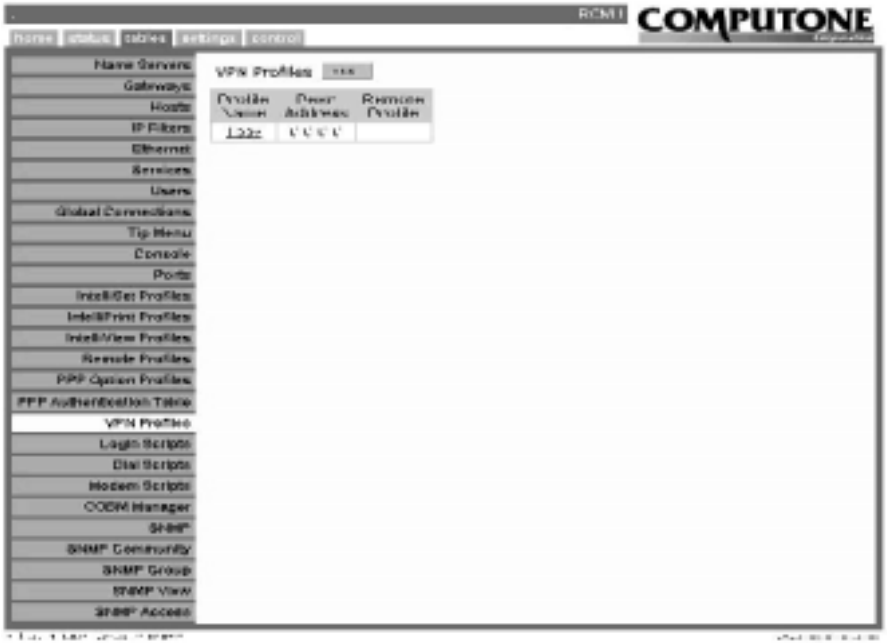


Figure 31 VPN Profiles Screen

Virtual Private Network (VPN) allows the virtual connection between a remote host (computer) and a private network. The connection is made through the network to a gateway machine on the private network which allows the remote computer to operate as if connected to the private network.A new *VPN* profile can be added or an existing profile can be configured.

To configure a profile, click on the **Profile Name** and the following screen appears:

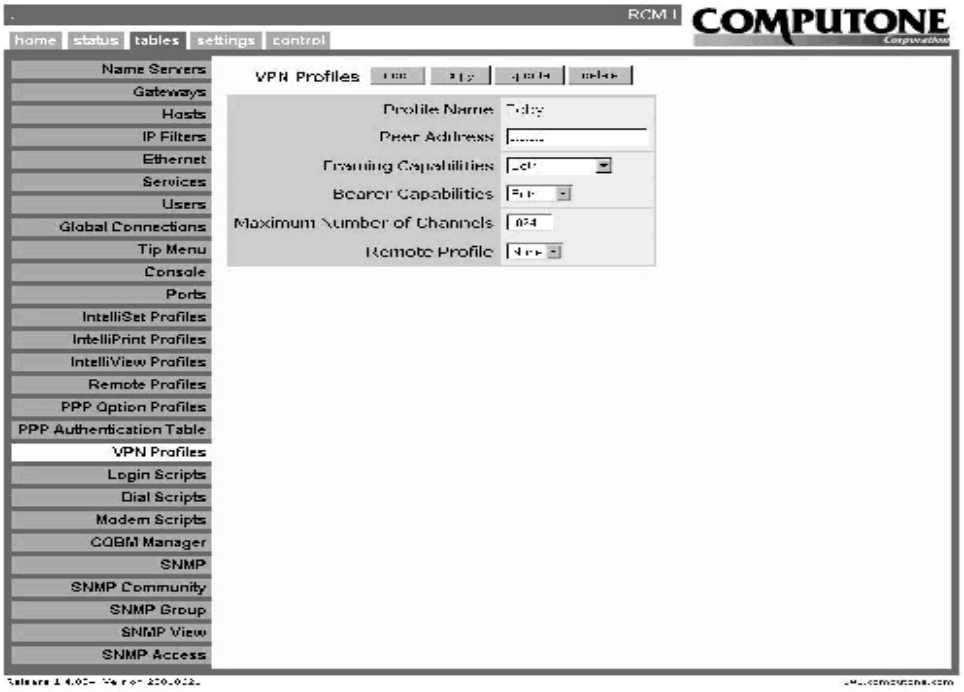


Figure 32 VPN Configuration Screen

The name and the address of the remote computer are displayed. The option fields are:

- ***Framing Capabilities- Asyn/Syn/Both*** provides point to point connection.
- ***Bearer Capabilities***- Digital/Analog connection.
- ***Maximum Number of Channels***- Set the number of channels.
- ***Remote Profile***-Predefined under ***PPP Options*** tab.

Login Scripts

Selecting the ***Login Scripts*** parameter displays the following screen;

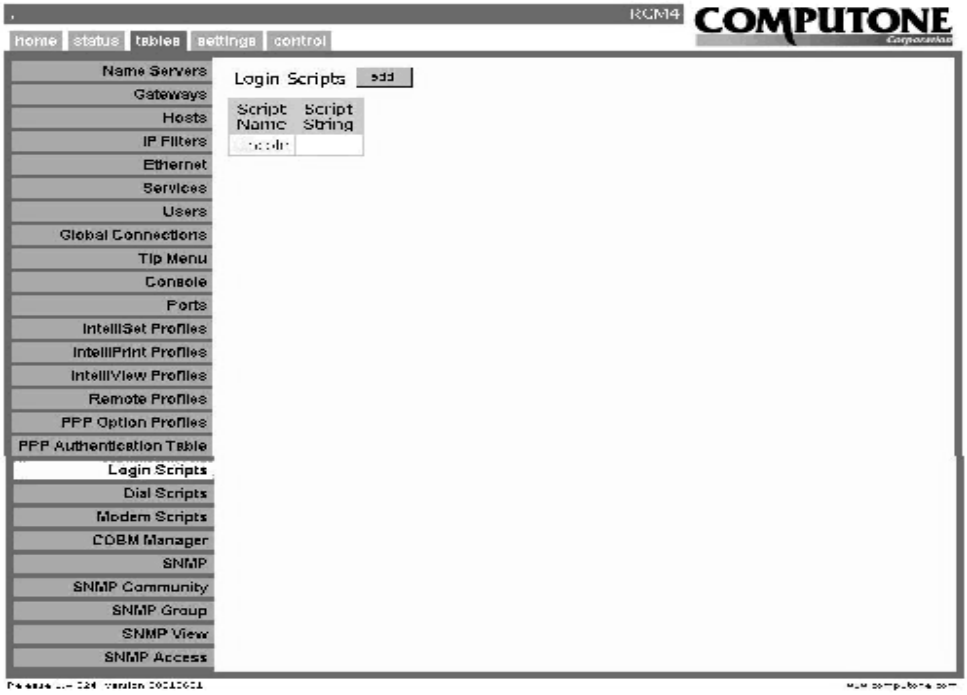


Figure 33 Login Scripts Screen

When the RCM 4/8 starts to bring up an outbound PPP or SLIP connection, it first uses the dialer script to make the modem dial the remote site. At the remote site, a modem answers the call and the host computer may be configured to issue a login or password prompt. Then, it brings up its side of the link (or hangs up) based on what user name and password are provided.

Login scripts are run immediately after the dial scripts, allowing the RCM 4/8 to provide automatically the necessary responses to a remote site's login, password, or other prompts. The nature of the login script is determined by the remote site that is contacted. Therefore, a login script is associated with a *remote profile*, not with a serial port (as are dial scripts).

Click on the script name and the ***Login Script String*** screen apprears.

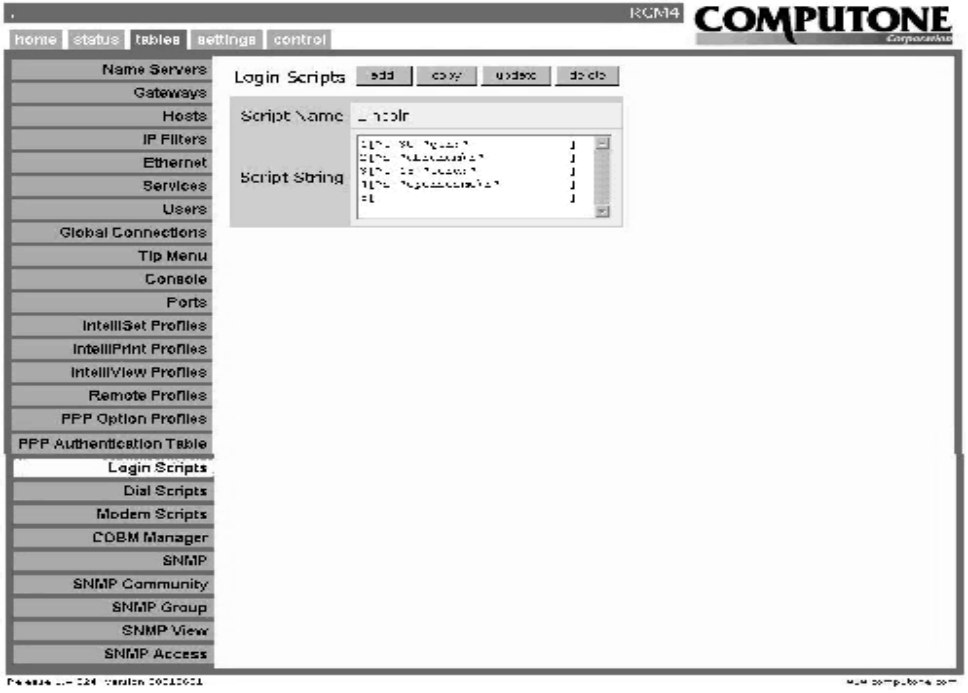


Figure 34 Login Script String Screen

The ***Script Name*** *(lincoln* in this example) is the unique name of this Login Script. During Remote Profile configuration, assign this to the profile's *Login Script*. The rest of the form contains the body of the script. Like Dial Scripts, it is composed of commands and the rules for forming these commands are the same as for Dial Scripts.

In the example shown in Figure 34 , the RCM 4/8 waits up to thirty seconds for data matching ***gin*:** to come in. Presumably, this indicates the remote host has prompted us for our login name. Then, the RCM 4/8 sends ***abraham***, our login name. Then, it waits for the password prompt, as indicated by the fragment ***word*:**. Finally, the RCM 4/8 sends our password, ***opensesme***. The login script is finished, and the RCM 4/8 brings up our side of the link and so does the remote site.

Dial Scripts

Selecting the ***Dial Scripts*** parameter displays the following screen:

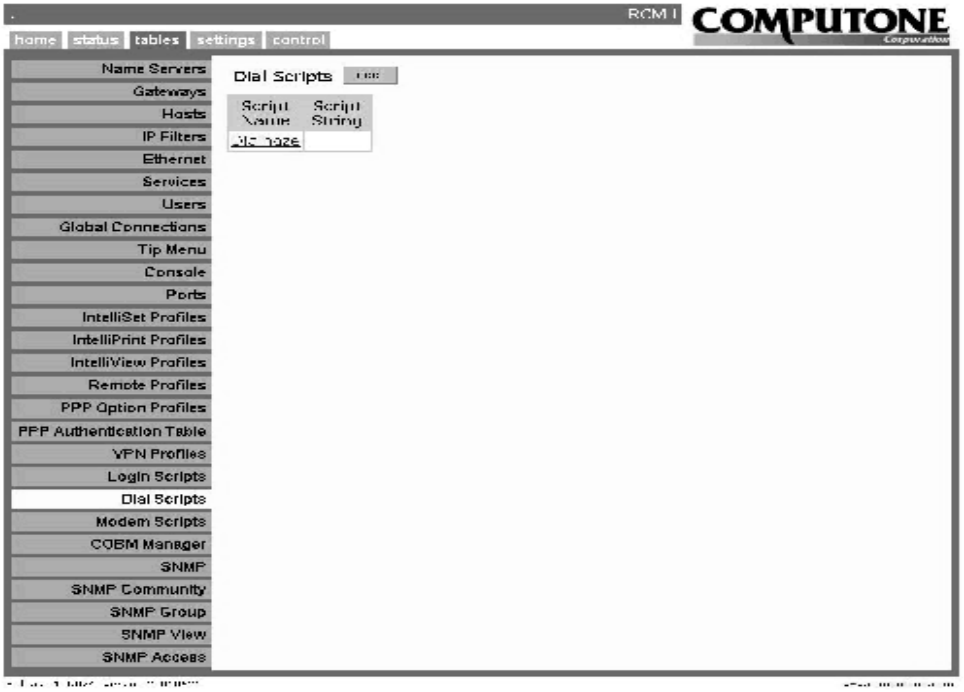


Figure 35 Dial Script Screen

New dial scripts can be added and an existing dial script configured. To configure an existing dial script, click on the dial script name and the ***Dial Script Configuration*** screen appears:>

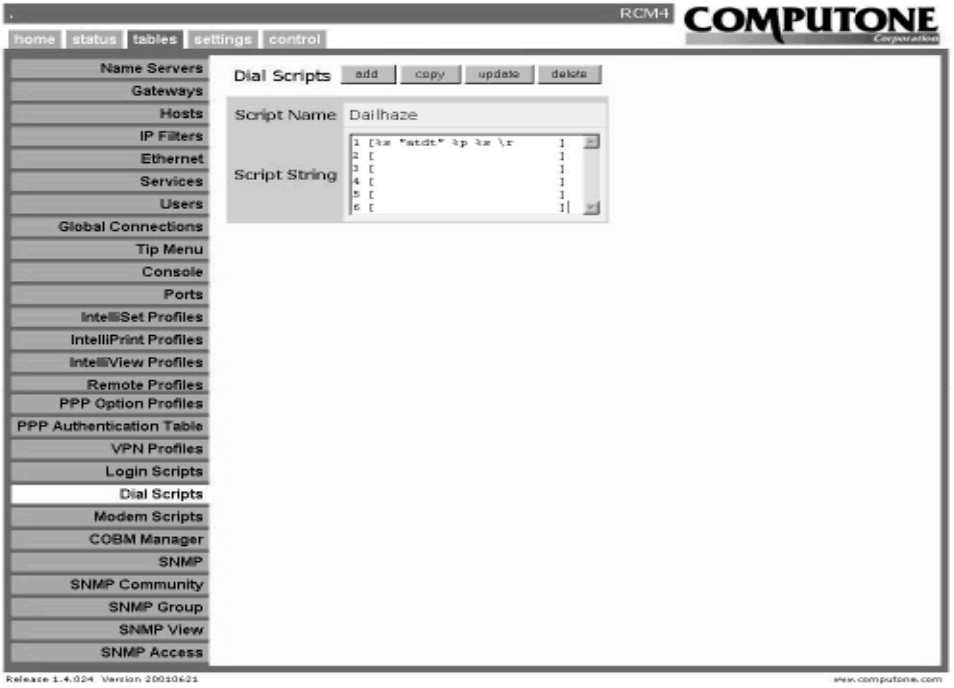


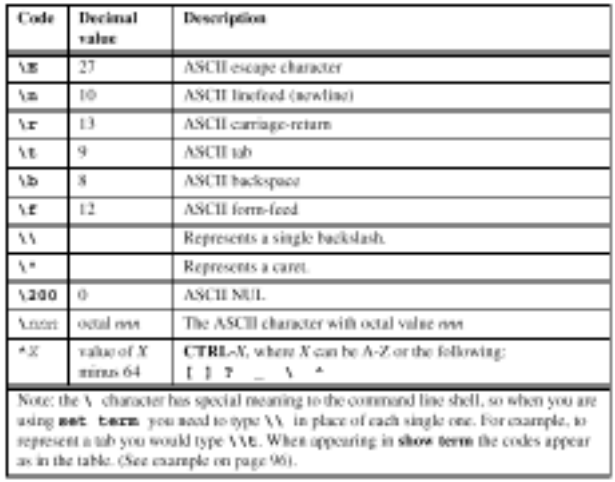
Figure 36 Dial Script Configuration Screen

Dial scripts define what needs to be sent to a modem so that it dials out and connects to another modem. The contents of the dial script depends somewhat on the modem. Since a given serial port is attached to a given modem, dial scripts are associated with serial ports. The ***Script Name*** *(dialhaze* in this example) is the unique name of this Dial Script. During serial port configuration, assign this to the port's *Dial Script.* The remainder of the form consists of six lines of forty-two columns each, but there is nothing special about the arrangement into rows and columns. Before the script is run, trailing blanks are removed from each line and all the lines are run together.

Not only must the script send out strings of data, it sometimes needs to wait until certain responses are *received* before continuing. Therefore, a Dial Script is built not from simple data strings, but from *script commands*. There may be several commands on a line, but a command must not be split across lines.

The following table shows how script commands are constructed:

Table 11 Dial and Login Script Commands

Command Definition / Examples	Description
%s `string` %s “ATDT5551212\r” %s “hello there” %s hellothere	Transmit the string to the serial port. If the string contains any spaces, enclose the entire string in quotes. Control characters can be represented by the following codes: 
%w `time string` %w 10 connect\r %w 5 “carrier” **%w** `time` %w 10 **%w** `string` %w carrier\r %w “10” %w “1derful”	Wait until the specified string is received from the serial port, or the time (in seconds) elapse, whichever comes first. Either the time or the string may be omitted. • If the time is omitted the script will wait forever for the string. • If the string is omitted the script will wait the specified time unconditionally. • If a time and a string are *both* given, getting the string first is considered good. Timing out first is considered bad. **When** an interface is using a dial or login script to bring up a connection, if a *wait* command times out before the string is received, the connection attempt will be stopped and the line disconnected. If there is only one thing after the **%w** how does the script know whether it is supposed to be the time or the string? If it is a number, it is assumed to represent a time. Otherwise it is a string. If wanting to wait forever for a certain string *and the string is a number*, then enclose it in quotes so it won’t be mistaken for a time. Control characters are represented in these strings the same as for the **%s** command.
%p	Send the phone number stored in the associated Remote Profile. This command allows the same dial script to support several outbound connections with different phone numbers. Otherwise, separate dial scripts would have been needed.

Code	Decimal value	Description
\E	27	ASCII escape character
\n	10	ASCII linefeed (newline)
\r	13	ASCII carriage-return
\t	9	ASCII tab
\b	8	ASCII backspace
\f	12	ASCII form-feed
\\		Represents a single backslash.
\^		Represents a caret.
\000	0	ASCII NUL
\nnn	octal nnn	The ASCII character with octal value nnn
^X	value of X minus 64	**CTRL**-*N*, where *N* can be *A*-*Z* or the following: [ ] ? _ \ ^

Note: the \ character has special meaning to the command line shell, so when you are using **set term** you need to type \\ in place of each single one. For example, to represent a tab you would type \\t. When appearing in **show term** the codes appear as in the table. (See example on page 96).

Modem Scripts

Selecting the ***Modem Scripts*** parameter displays the following screen:

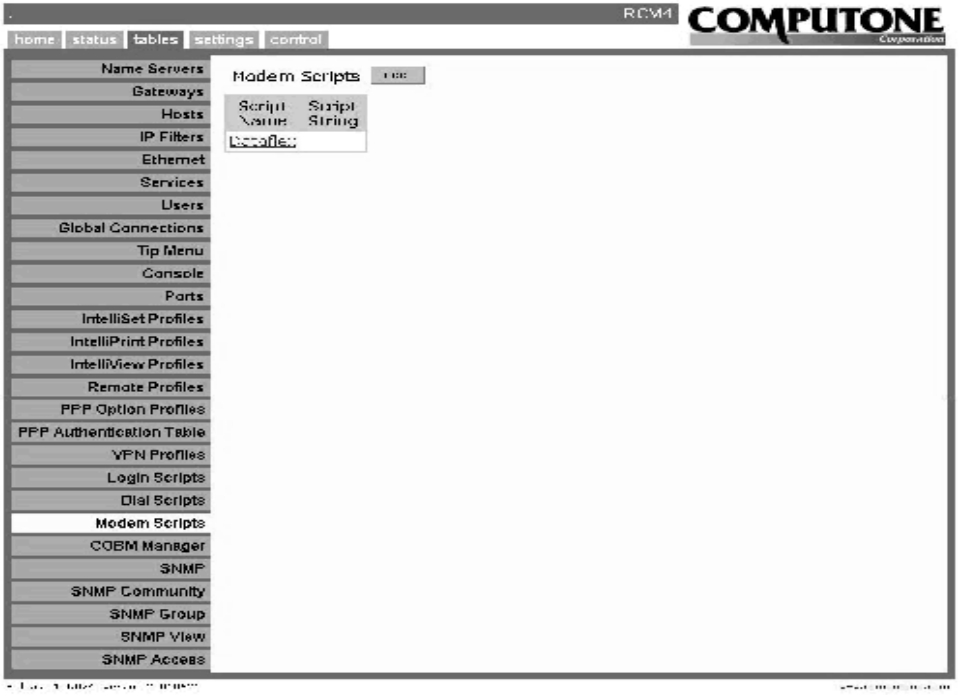


Figure 37 Modem Scripts Screen

To edit the script, click on the script name. The following screen is displayed.

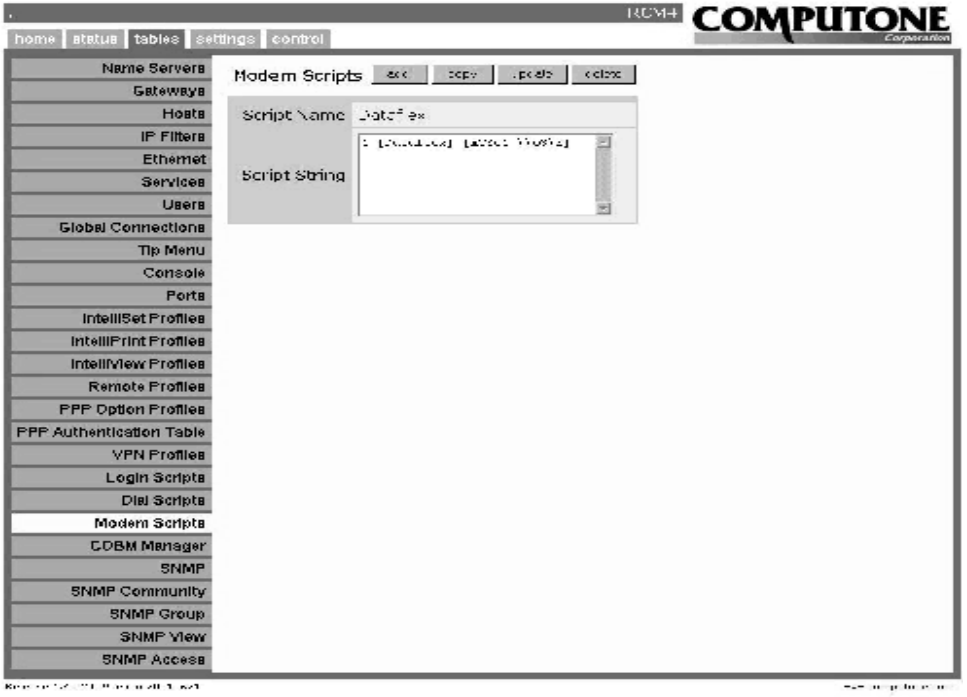


Figure 38 Modem Configuration Screen

In this example, entry 1 is named “Dataflex” and contains the following commands:

- ***AT*** - begins the command line.
- ***&C1***- only turn on DCD when remote carrier is present.
- ***\Q3*** - enable RTS/CTS flow control. Although the command contains a single backslash, enter it twice in the table because backslashes are used to introduce special characters (see chapter 5, *Character Codes in Strings)*.
- ***\r*** - carriage return or end of command. Note the use of the backslash to start a sequence that represents a control character.

If a port is configured to have a modem initialization string “Hayes”, then the port would send ***AT&C1\Q3*** (***return)*** as the command. There is nothing special about these particular commands, they were chosen for a sample only. To remove

an entry from the table, select the input area for the name and press ***Ctrl-Z*** to erase it or replace it with a different entry.

Be careful when removing an entry because there is no check to make sure it wasn't being used. If the RCM 4/8 can't find an entry in the table whose name matches the *modem init* specified for that port, it assumes what configuring was the initialization string itself, not the name of one.

COBM

Selecting the ***COBM*** parameter displays the following screen:

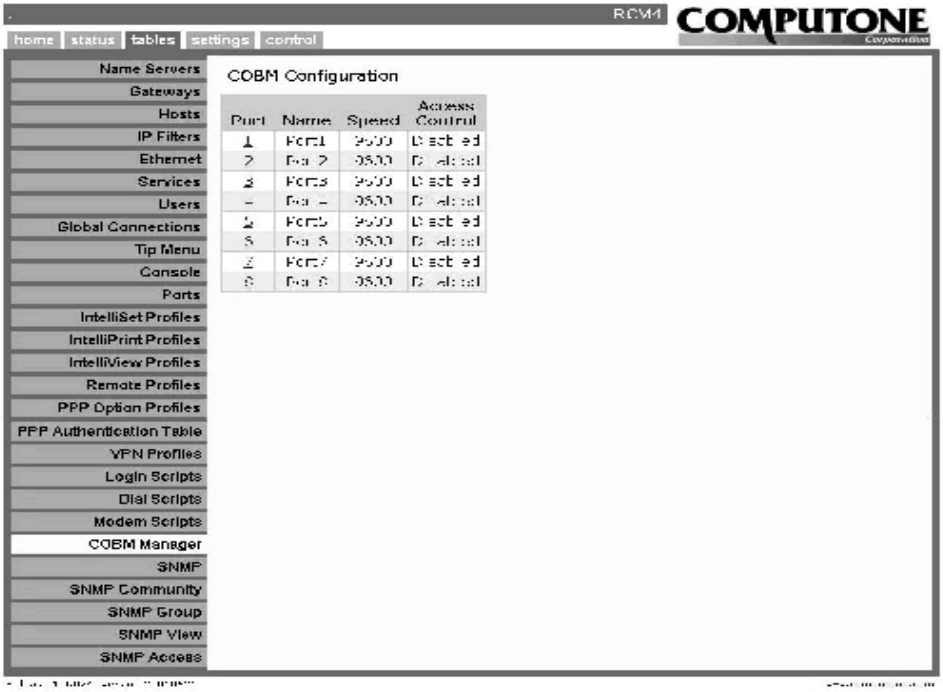


Figure 39 COBM Screen

Overview

The COBM or *Computone Out-of-Band Management* can be enabled/disabled via ***Application*** under the setting tab. When a workstation requests a connection to the COBM manager will perform a reverse lookup of the IP address of the

workstation making the request. The manger must be able to resolve the name in the local host table or the DNS name server. If the name is not verified, the connection request will be rejected.

To configure a port, click on the port number, and the following screen appears.

Figure 40 COBM Configuration Screen

The ***Name*** field is the text screen name of the machine/server that is connected to the port. The options to configure are:

- ***Speed*** - Set to match the speed of the connected device.
- ***Bits***, ***Parity***, ***Input Flow***, ***Output Flow*** and ***Stop*** - All set to the requirements of the attached device.

- ***Access Control*** - The choices are ***Disable*** (not COBM managed), ***Private***, and ***Full*** (COBM managed). If ***Private*** is set, the host/network address configured on this port, only apply to this port. In most cases ***Full*** is the desired selection, which allows any host in the ***Reject***, ***Allow***, and ***Trust*** fields to accumulate and apply to any port.
- ***Alert Trigger*** - The text screen incoming data on the port monitored for a match on a trigger. If a match is found the COBM manager will alert the COBM workstation. The workstation is only alerted if being monitored by the IntelliServer.
- ***Reject Hosts*** - IP address that are prevented from connecting to the COBM list.
- ***Allow Hosts*** - IP address that are allowed to connect to the COBM list. Host will be required to enter a password when requesting a connection. As a result, the host must have an account on the IntelliServer.
- ***Trust Hosts*** - All Ip address listed will be given access without a password.

SNMP

Selecting the ***SNMP*** parameter displays the following screen**:**

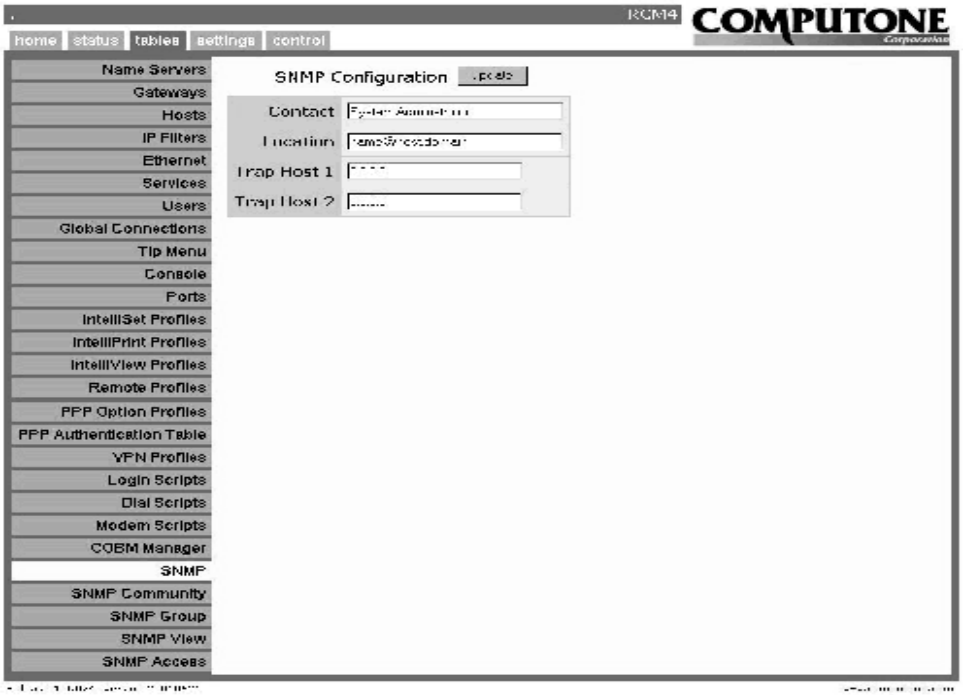


Figure 41 SNMP Configuration Screen

Overview

SNMP, or *Simple Network Management Protocol*, requires the following:

- One or more SNMP *managers*. A manager is a network computer that is running one or more SNMP management applications.

One or more SNMP *agents*. Agents are network computers and devices (such as the RCM 4/8) that can respond to queries from SNMP managers. The SNMP managers use UDP datagrams to send commands and queries to the agents and the agents send back responses (also using UDP). Agents also can send unsolicited messages, called *traps*, to report important conditions such as shut-down and start-up. The hosts that receive these traps are called *trap hosts.*

Trap Hosts

The RCM 4/8 can send *trap* messages to as many as two *trap hosts*, which can be configured using the configuration screen. When using the configuration screen, enter IP addresses in either or both of the spaces provided. If there is only a single trap host, SNMP Trap Host2 is set to **0.0.0.0.**

Type in the desired IP address to **add** a new trap host or **delete** an existing one. A new trap host cannot be added if there are already two configured. One must be deleted first because this changes the IP address for that host to 0.0.0.0. To change an existing entry, delete it first and then add a new one.

Table 12 SNMP Screen Configurations

Selection	Description
Contact	Name of person who supports this function.
Location	Physical location of this machine.
Trap Host 1	IP address of Trap Host 1. A *Trap Host* is a host that sends unsolicited messages, called *traps*, to report important conditions such as shutdown and start-up.
Trap Host 2	IP address of Trap Host 2.

Selecting the ***SNMP Community*** parameter displays the following screen**:**

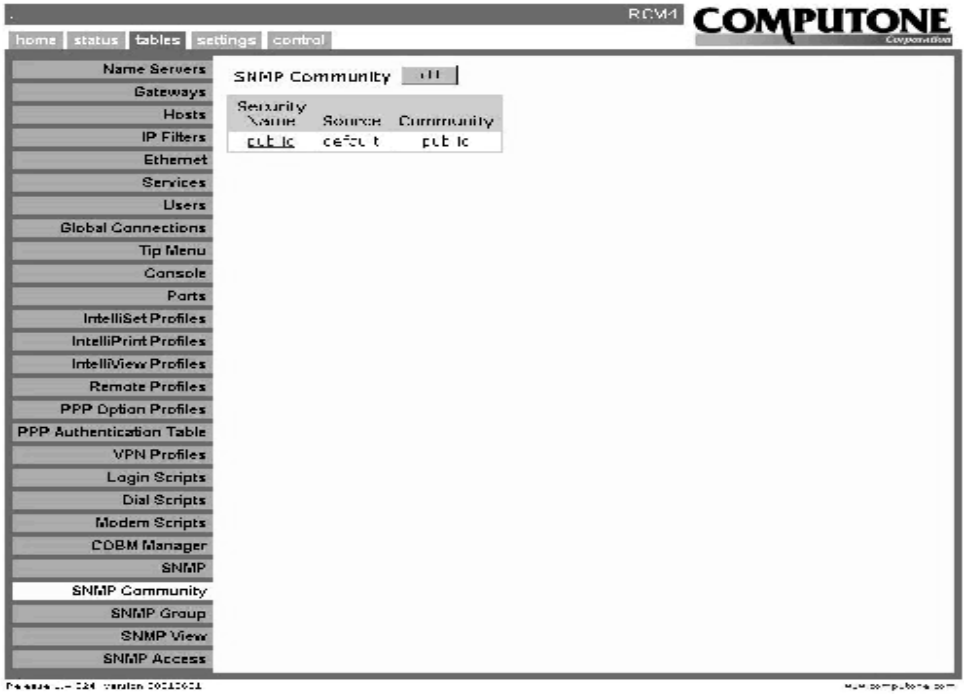


Figure 42 SNMP Community Screen

New Security Names can be added to the SNMP Community by clicking the ***add*** button. Click on a ***Security Name*** and the following screen appears:

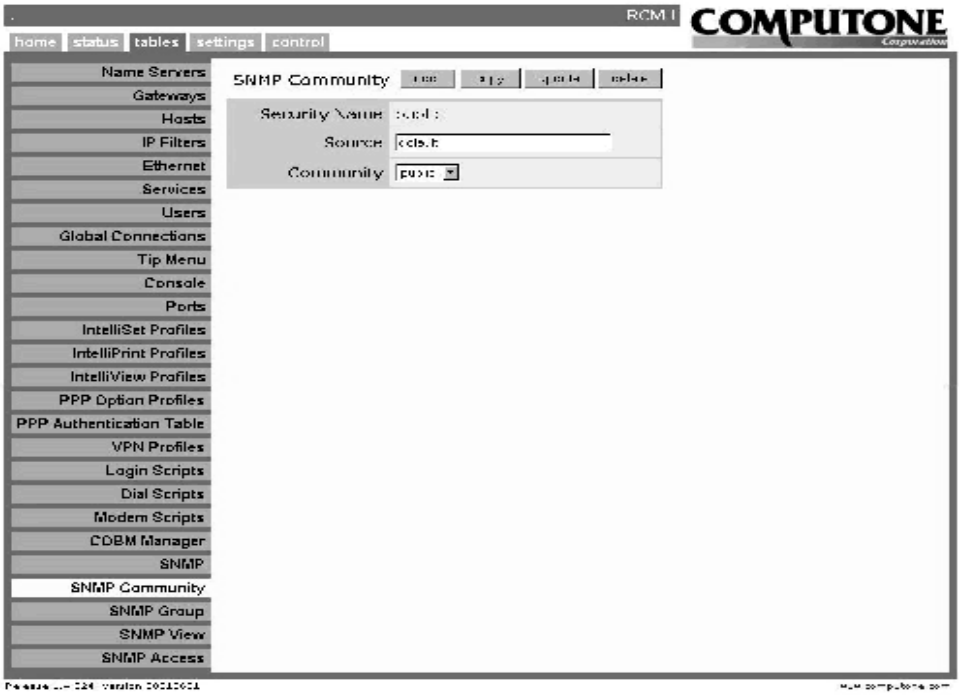


Figure 43 SNMP Community Configuration Screen

The ***Source*** field can be a host name, subnet or default, and the Community can be defined as public or private. When changing the information, make sure the ***update*** button has been pressed to enter the new information.

Selecting the ***SNMP Group*** parameter displays the following screen**:**

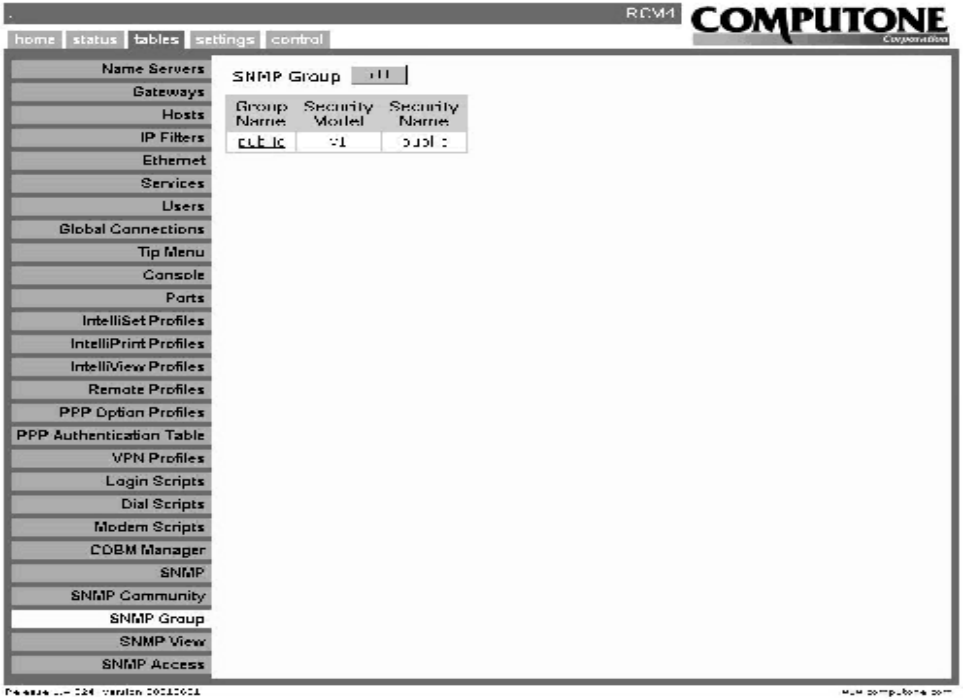


Figure 44 SNMP Group Screen

Groups belong to a community.To add a new group name, click the ***add*** button.

Configure the group by clicking on the ***Group Name***, and the ***SNMP Configuration*** screen appears.

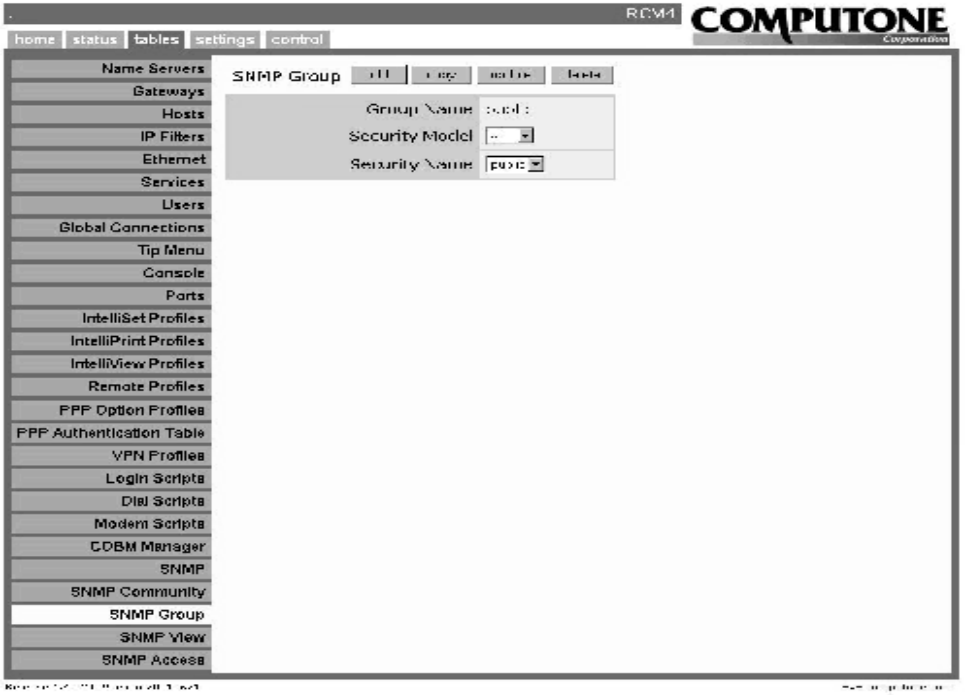


Figure 45 SNMP Group Configuration Screen

The options for the **Group Name** are:

- ***Security Model*** - Choices are ***V1***, ***V2c*** and ***USM***.
- ***Security Name*** - Choices are ***Public*** and ***Private***.

Selecting the ***SNMP View*** parameter displays the following screen**:**

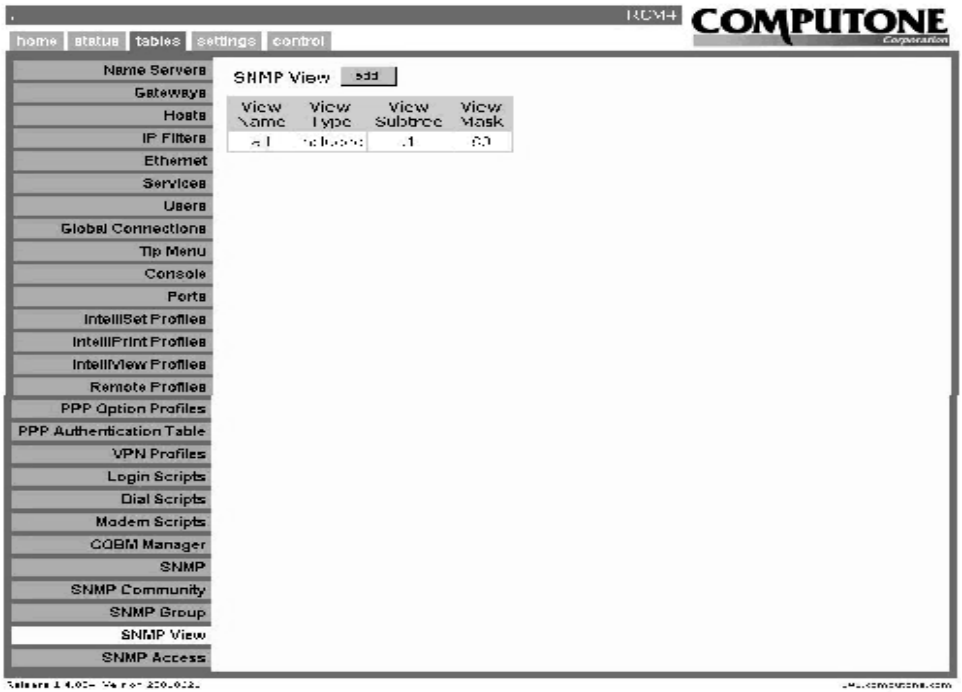


Figure 46 SNMP View Screen

Allows the viewing of all or any ***View Names***.

To configure a name, click on the ***View Name*** and the ***SNMP View Configuration*** screen appears

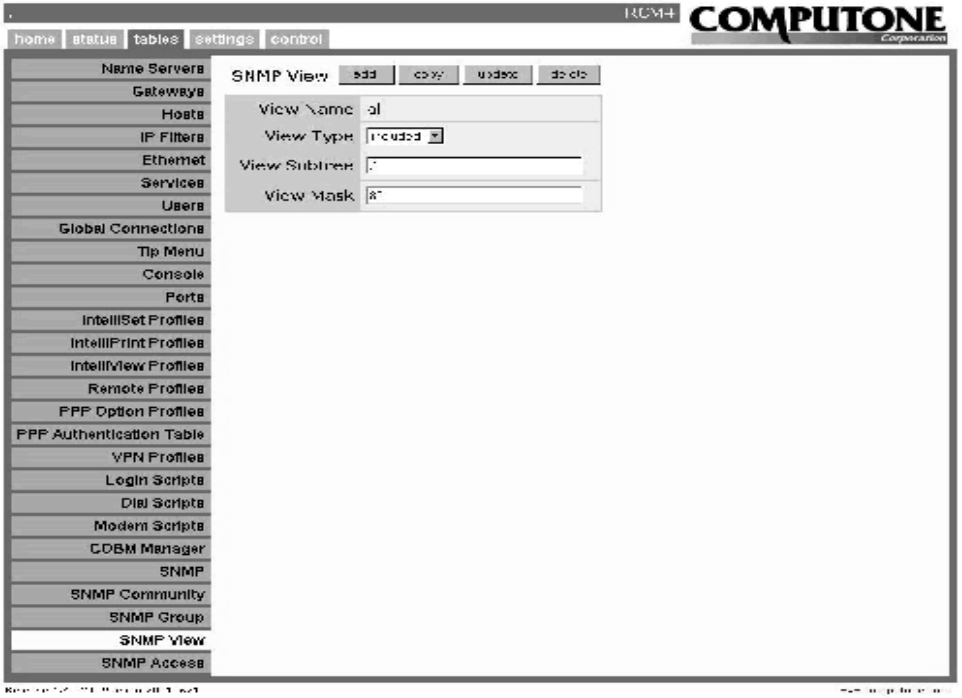


Figure 47 SNMP View Configuration Screen

The ***View Type*** can be configured as ***included*** or ***excluded***.

Selecting the ***SNMP Access*** parameter displays the following screen**:**

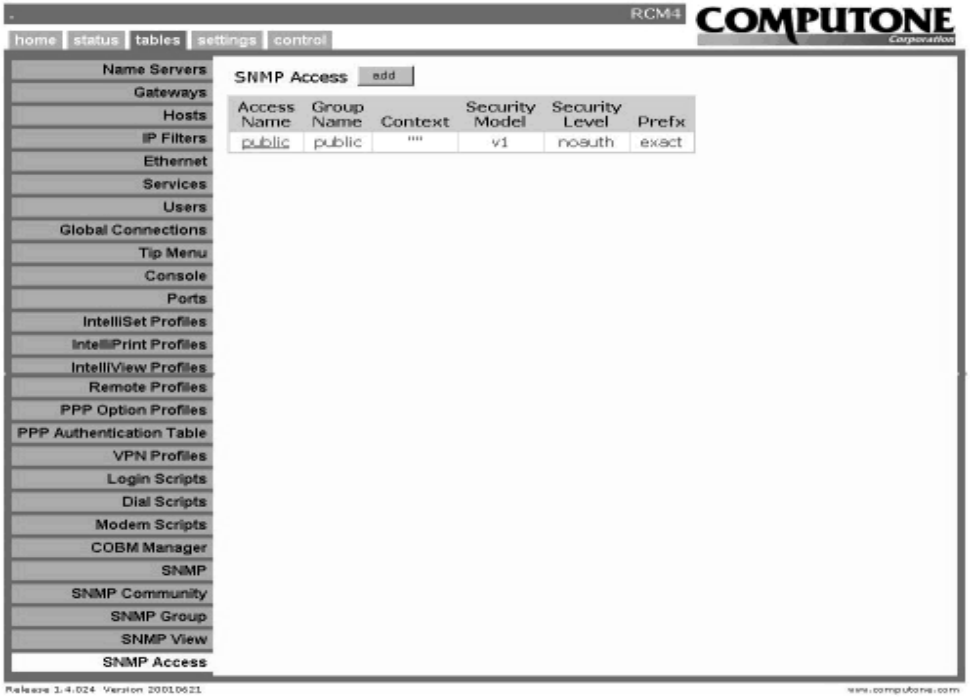


Figure 48 SNMP Access Screen

To add an **Access Name**, click ***add*** button.

To configure an ***Access Name***, click on the name. The ***Access Configuration*** screen appears:

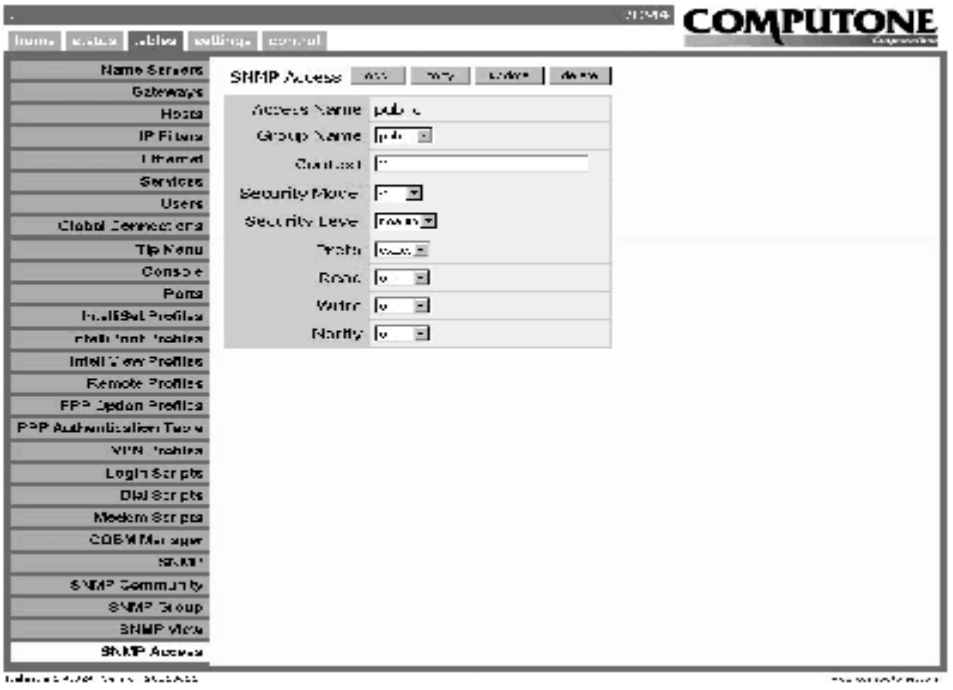


Figure 49 Access Configuration Screen

The options available to configure are:

- ***Group Name*** - Select the group name from those available.
- ***Security Model*** - The choices are ***Any***, ***V1***, ***V2c***, and ***USM***. The selection must be the same as selected in the group security model. **Any** works with all groups.
- ***Security Level*** - The choices are ***noauth***, ***auth***, and ***priv.***
- ***Read***, ***Write*** and ***Notify*** - The choices are ***all*** and previously set names from the ***View Name*****s**.

CHAPTER 4

Configuring System Settings

The following table lists the selections available for configuration through the ***Settings*** tab:

Table 1 Settings Parameters

Parameters	Description
System	Contains the host name, domain name, IP address, Ethernet address, console port, IP filter, RIP type, login prompt, password prompt, and user prompt.
Applications	Provides selection for Web Server (httpd), Secure Shell (sshd) and Insecure Shell (telnetd).
Boot	Provides selection of Boot Type, Host 1, File 1, Host 2, File 2, and Retry Count.
Syslog	Provides selection of Syslog Host, Syslog Facility, and Syslog Priority.
RADIUS Authentication	Use to configure Remote Authentication Dial-In User Service (RADIUS).
RADIUS Accounting	Use to configure RADIUS Accounting parameters.
RIP	Provides selection of State, Version, Domain (RIP-II Only), Hot List Type, Host 1, Host 2, Host 3, Host 4, and Password (RIP-II Only).
Secured Shell	Provides selection of Hot Key Size, Server Key Size, Authentication Grace Time, Server Key Regen, TCP Port, Authentication Method, and Allow Root Login.

Settings Tab

Selecting the ***Settings*** tab displays the following screen:

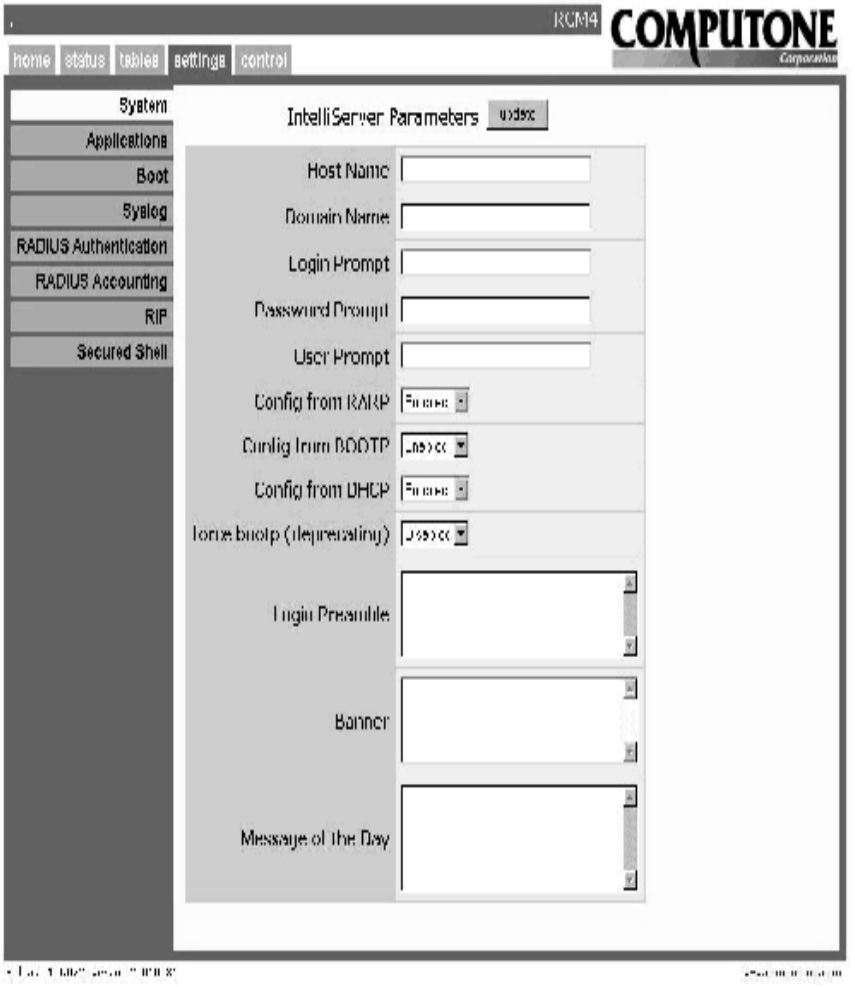


Figure 1 System Screen

The following table defines the parameters on the ***System*** screen.

Table 2 System Parameters

Menu Entry	Description
Host Name	Although Internet Protocol identifies hosts and networks by their IP addresses, these addresses are not very practical for a human being to use.To remember that Computone's FTP site was 160.77.1.10 and Generic General's WEB page was on 160.77.99.101, and so on, it would get difficult very quickly. That is why it is possible to identify a host by a name, rather than by its IP address. Enter the host's name here.
Domain Name	Once given a registered Internet addresses, a registered "domain name" can be requested. This domain name needs to be added to the end of any host name that has been assigned. Domain names are organized in a hierarchic structure. The universe of names the "root domain") has been divided into basic groups, each with its own domain name and each potentially with its own administrator. For example, the domain**.com** assigns domains to commercial networks,**.gov** to government agencies, **.edu** to educational institutions, and **.org** to non-profit organizations. Notice that the individual elements of a domain name are separated by periods. Computone is a commercial establishment, so it has a domain name from whoever administers the **.com** domains.
Login Prompt	Allows changes in the login prompt.
Password Prompt	Allows changes in the password prompt.
User Prompt	Allows changes in the user prompt. **NOTE:** If logged in as user root, the user prompt is overwritten by a # sign.
Config from RARP	Enables or disables the ability of the unit to try to configure itself (MAC address, name, and so on) on power-up from a RARP server.
Config from BOOTP	Enables or disables the ability of the unit to try to configure itself (MAC address, name, gateway to use, nameserver to use, and so on) on power-up from a BOOTP server.
Config from DHCP	Enables or disables the ability of the unit to try to configure itself (MAC address, name, gateway to use, nameserver to use, and so on) on power-up from a DHCP server.

Table 2 System Parameters

Menu Entry	Description
Force bootp (deprecating)	If this is selected, the unit tries to configure itself from bootp and uses the information stored there first. If there is not enough information available, it then searches the network for a bootp server.
Login Preamble	The preamble is sent before the user logs in. Information can be provided for the user such as login name to use, whether a password is required, and so on).
Banner	The login banner contains two lines: the first line always says "IntelliServer Release" followed by the release number of its software. The next line has the IntelliServer's *node name* (a.k.a. its host name) followed by a login prompt and is sent after the user logs in.
Message of the Day	The message-of-the-day is sent after the user logs in. Enter information you want to user to see for today. For instance, "The network will be down from 1-3 PM today."

Applications

Selecting the ***Applications*** parameter displays the following screen:

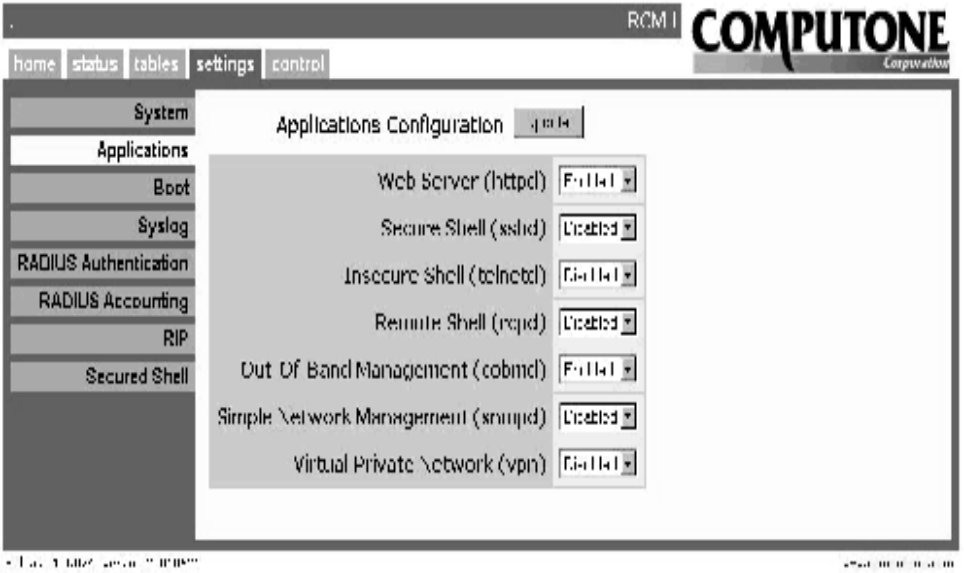


Figure 2 Applications Screen

This screen allows you to enable or disable the web server (httpd), secure shell (sshd), insecure shell (telnetd), remote shell (rcpd), out of band management (Cobmd), simple network management (snmpd) or virtual private network (vpn).

IMPORTANT: If the web server is disabled from this screen, save the configuration, and re-enable it from the command line by entering the following command if desired:

```
apps set httpd enable
```

It is possible to disable the web interface, secure shell and insecure shell all at the same time. If a port has been previously defined as login-by-port, undo this situation by accessing this port. Otherwise, factory defaults must be restored and the RCM 4/8 then reconfigured.

There are three ways to restore factory defaults to NVRAM:

1. Enter *restore factory* at the shell command prompt if you can access the command line.
2. Hold the paperclip button down while powering-up the RCM 4/8. Release it after 1 second.
3. Tap the ESC key a few times while powering-up the RCM 4/8.

Even if method 2 or 3 is used, it is recommended that a terminal or PC be attached to the console port. The reason is that the machine has no IP address and, thus, is not accessible over the network. None of the methods above alter NVRAM so if the old configuration is desired, do a *restore*, *shutdown now*, or a power cycle. If the factory defaults are desired, type in *save* as if saving any other configuration.

Figure 3 shows the location of the paperclip button to restore factory defaults.

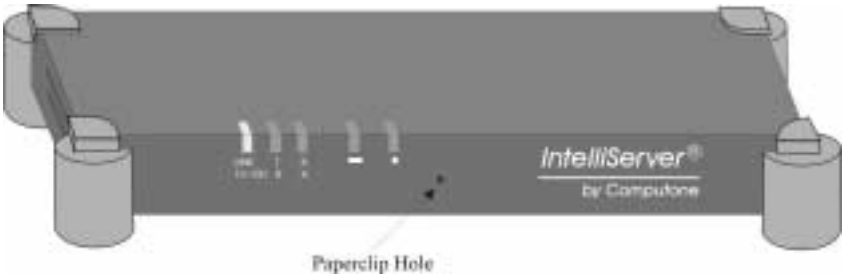


Figure 3 Restoring Factory Defaults

Boot

Selecting the ***Boot*** parameter displays the following screen:

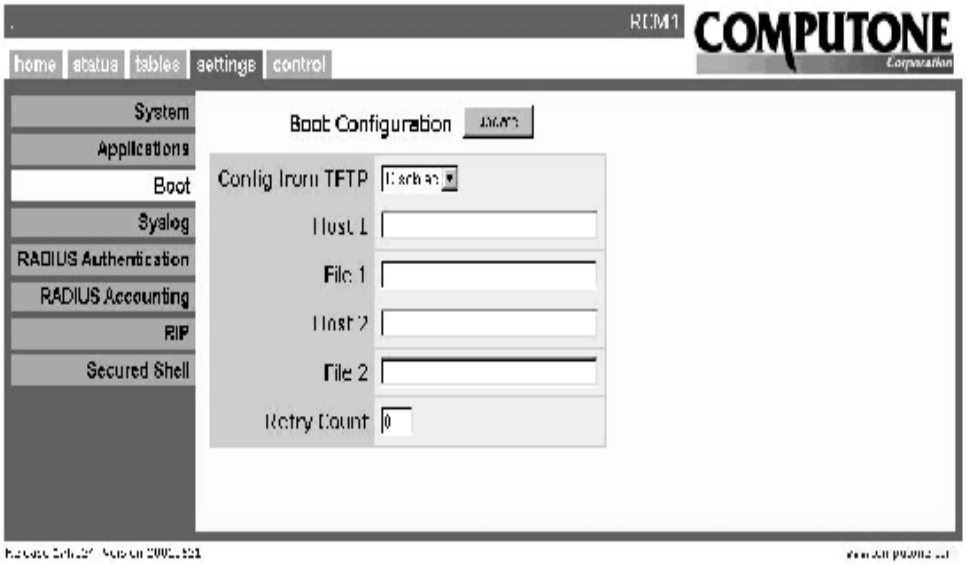


Figure 4 Boot Screen

The RCM 4/8 has the option of running the version of software stored in its internal PROM, or of running a later version stored on one of the hosts on the local network. Booting a newer software version over the network is the most common method of upgrading when new releases of RCM 4/8 software become available. The RCM 4/8 also has the option of using the configurations stored in internal NVRAM, or of using configurations stored in a file on one of the network's hosts.

This option determines whether the RCM 4/8 tries to get its software and con-figuration information from the network. Table 3 defines the boot type selections.

Table 3 Boot Configuration Parameters

Parameter	Description
Boot Type	
Disabled	The RCM 4/8 runs using the software stored in PROM and the configuration stored in its local NVRAM. It does not attempt to get any of this information over the network.
Enabled	The RCM 4/8 uses TFTP to access a host on the network to download configuration information. Primary and secondary TFTP hosts, boot files, and configuration files need to be specified.
TFTP	The RCM 4/8 uses TFTP protocol to download a *TFTP Boot File* and *TFTP Config File* from a *TFTP Host. Primary* and *Secondary* files and hosts may be specified, and if the primary fails, the secondary is used.
(Primary) Host 1	IP address of the host to check first for a configuration file.
(Configuration) File 1	Name of the configuration file on Host 1.
(Secondary) Host 2	IP address of the host to check second for a configuration file.
(Configuration) File 2	Name of the configuration file on Host 2.
Retry Count	This controls the number of times the RCM 4/8 attempts to boot from the network before it gives up and uses the software and configuration stored locally. If the retry count is set to **0,** then it continues to retry forever.

When the RCM 4/8 is configured to netboot, it first must bring up its own software in order to start up the networking code so that it can do the netbooting. When watching the console, the older version's messages and banners are viewed. Since it knows it must netboot, the RCM 4/8 configures itself to allow space in DRAM to download the new software; most serial ports are deactivated and non-essential processes are removed. After it is loaded, the new software is started and while watching the console *its* power-up messages and banners are viewed, which look almost like the first set, and then the RCM 4/8 is running.

Primary TFTP Host and Config File

These are used when the *Boot Type* is set to TFTP. Host 1 (Primary TFTP Boot Host*)* is the IP address of the first host the RCM 4/8 tries to download its files from. File 1 (*Primary TFTP Config File* contains a configuration file that had earlier been saved from a RCM 4/8 to this host.

When Net-booting Fails

If the netbooting should fail after a predetermined number of retries, it finally brings itself up using the software and configuration in PROM and local NVRAM. To do this, it cannot simply stop trying to TFTP the files. It has to actually reboot itself again, because it had previously reconfigured its DRAM for net-booting. This means temporarily deleting things which it now must reload from PROM to recover. The net result is that the console displays a double set of banners in this case as well. A RCM 4/8 which has not yet been configured with an IP address also uses BOOTP protocol in order to learn this and other information. This happens regardless of the **boot type** settings.

Syslog

Selecting the ***Syslog*** parameter displays the following screen:

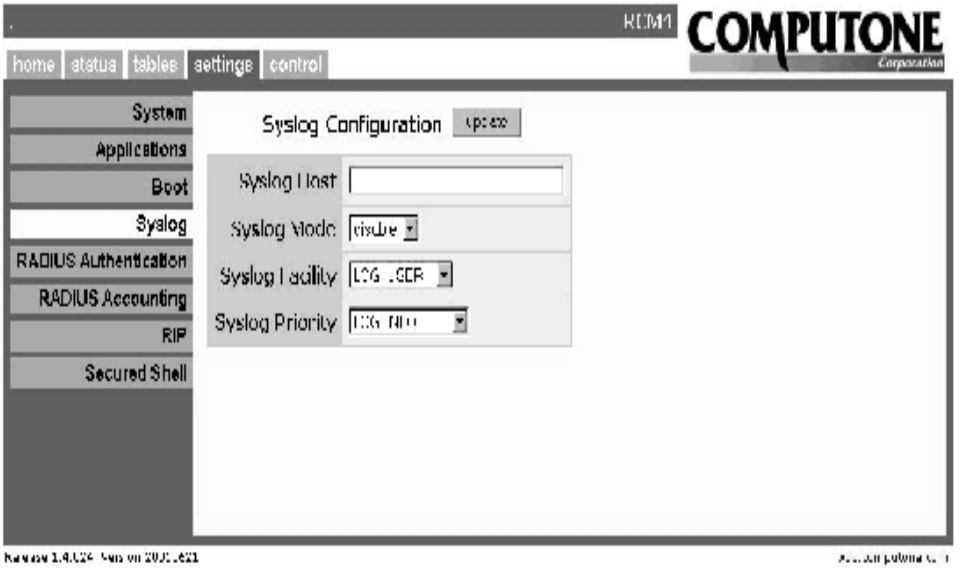


Figure 5 Syslog Screen

Network syslogging is a way of keeping track of what is going on within the system. There are two parts to syslogging:

1. The *syslog host* - a computer on the network running software called a system log daemon (on UNIX hosts usually called *syslogd*).
2. One or more *syslog clients* - computers and devices configured to send syslog messages to the syslog host.

Syslog messages are sent as UDP datagrams and the syslog host does not confirm receipt by returning any acknowledgment. Therefore, syslog messages are intrinsically unreliable. While in most networks most syslog messages will be delivered most of the time, it is still possible that a message can be lost. Messages are especially apt to be lost if an extremely large number are sent to the same host very quickly, and especially if that host is otherwise busy.

Each syslog message contains three parts:

Table 4 Syslog Message Components

Syslog Message Components	Description
Message Text	The message itself. By convention this message begins with something to indicate which of the sender's processes generated the message. That is, messages generated from the *init* process might be expected to begin with *"init:"*, for example.
Priority	This identifies the urgency of the message. Syslog clients can be configured to send only messages of a certain priority or higher. Syslog hosts can be configured to store messages based on priority: messages of a certain urgency and higher being sent to a certain file, messages of a certain priority or lower are discarded, and still others are recorded elsewhere.
Facility	The facility serves to classify the source of the message. In that way, messages from user processes on the computer can be distinguished from system messages and other types. The syslog daemon can be configured to record messages from different facilities in different files. the separation that results for messages with different priorities. When messages of a certain priority are sent to a file, any messages with greater priority would also be sent to that file.

There are two more pieces of information that the syslog host will usually record in the log files:

1. The source (IP address or host name) of the syslog message. The IP address is obtained from the message's packet header.
2. The time the message was received. This is obtained from the syslog host's resident clock.

Syslogging provides an important debugging tool in many situations, so you should be comfortable with it. On the RCM 4/8 it is especially useful when there is trouble bringing up a PPP connection, and when there is a problem involving Reverse-TCP ports.

Table 5 explains the Syslog screen selections.

Table 5 Syslog Screen Selections

Parameter	Description
Syslog Host	This is the IP address of some host on your network which is configured to receive *syslog* messages.
Syslog Mode	
disable	Disables syslogging.
console	Syslogging is to console.
host	Syslogging is to host specified.
Syslog Facility	When the RCM 4/8 sends syslog messages to a syslog host, it identifies them by *facility.* Ultimately, this controls how the syslog host will log these messages: the syslog host will have been configured to record messages from some facilities into one file, and from other facilities into another.
LOG_USER	The facility is **LOG_USER** by default, but you can set it to any of the following: LOG_LOCAL1, LOG_LOCAL2, LOG_LOCAL3, LOG_LOCAL4, LOG_LOCAL5, LOG_LOCAL6, LOG_LOCAL7. There will probably be processes running on your syslog host which also syslog at the **LOG_USER** facility, so leave the RCM 4/8's facility at the default if all the messages are to be logged together. If your RCM 4/8 has been configured to generate lots of syslog messages, then a file all their own may be created. In this case, set it to one of the other facilities that no one else is using and see that your syslog host is configured to put those messages in a separate file.

RADIUS Authentication

Selecting the ***RADIUS Authentication*** parameter displays the following screen:

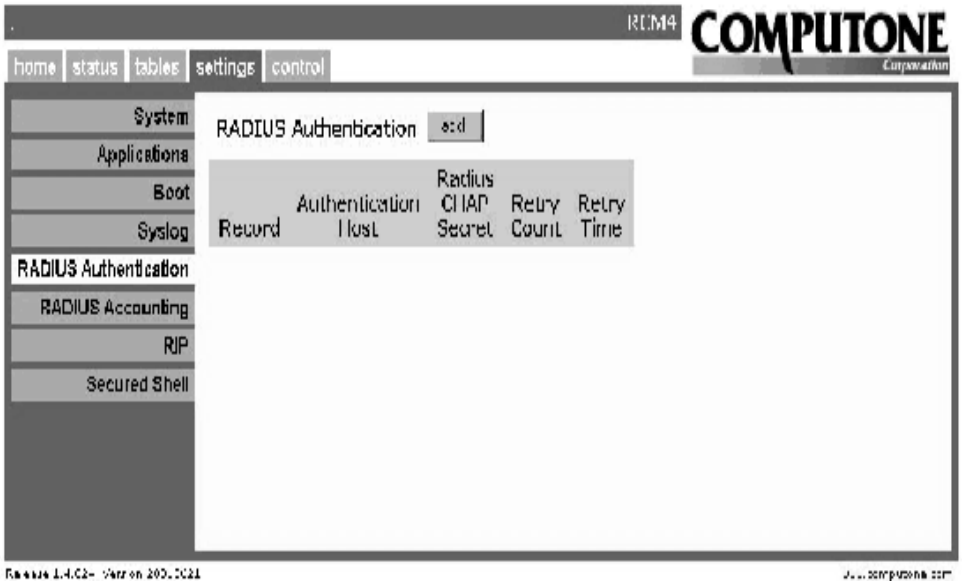


Figure 6 RADIUS Authentication Screen

The following table describes the ***RADIUS Authentication*** screen entries.

Table 6 RADIUS Authentication Screen Entries

Entries	Description
Radius Host 1	Host name of main RADIUS authorization server or its IP address.
Radius Host 2	Host name of alternate RADIUS authorization server or its IP address.
Radius CHAP Secret	Authenticates authorization requests to RADIUS Server.
Retry Count	The number of times the RCM 4/8 sends an authentication request to a host and waits for a reply. If there is no reply this request is re-sent a few times, and if a secondary RADIUS host is defined, it is tried as well. If there is still no reply, or if one of the replies is an *access rejection*, the connection is dropped.
Retry Time	RADIUS retry time in seconds.

The following table defines different elements of a *RADIUS* system.

Table 7 RADIUS Elements

Elements	Description
RADIUS Server	On some host, somewhere on your network, there is installed a software package known as a "**RADIUS server**". This package includes configuration files that have a list of users and their associated configuration, and a means for you to create and maintain this list. There will be a 'daemon" program which runs in the background and listens on the network for authentication requests from "**RADIUS clients**" (including RCM 4/8s). There will also be configuration files that control which clients this RADIUS server is authorized to respond to, and additional security keys to ensure that the requests are actually coming from the authorized source.
Three RADIUS Elements	Regardless of the software implementation, there will always be: 1. The RADIUS Server software (including the RADIUS daemon). 2. A user authentication file and some means to maintain it. 3.A list of authorized clients, with associated security keys.
Logging into a Client	When a user tries to log in to a client, it sends authentication requests to the RADIUS server. When the RADIUS server gets the request, it looks up the user's information, and sends a reply back to the client. What information does the client need for this: 1. The IP address or host name of the RADIUS server. 2. The security key to be used.

Table 7 RADIUS Elements

Elements	Description
Two Parts of RADIUS	The two parts of RADIUS are: 1. RADIUS Authentication. 2. RADIUS Accounting. RADIUS Authentication occurs when a user tries to log into the RADIUS *client*. After prompting the user for login name and password, the client sends this information in an *authentication request* to the RADIUS server. The RADIUS server checks the validity of the request, then checks its database of user names and passwords. If they are bad it sends a *rejection* back to the client, which in turn rejects the login. If the login name and password are good, the RADIUS server sends back a packet containing information about this user and the client (i.e., the RCM 4/8) uses this information to decide what type of service to supply for the user. RADIUS Accounting occurs when a user logs into or out of a RADIUS *client* after approving the login (either through an internal database or through RADIUS authentication). The client sends notification to the accounting server that this particular user has logged in. When the user logs off or is disconnected, the client also sends notification including the number of seconds the user was connected. When the RADIUS Accounting server receives these notices, it stores the information and then sends an acknowledgment back to the *client*. If the client does not receive an acknowledgment for its notices, it assumes they were lost and sends out duplicates. **RADIUS authentication can be done without doing accounting, or accounting without authentication.** If doing both, the accounting server can be the same host or a different one from the authentication server. Secondary authentication and accounting hosts can also be defined which the RCM 4/8 uses when there is no reply from the primary servers.
Things to Configure	Regardless of the software implementation, you must configure the following: 1. A list of authorized clients and their shared secrets. The RADIUS server needs to know the IP addresses of all the authorized RADIUS clients. Along with each client's address is a *secret*. You can pick whatever you like, but this same secret has to be configured into the *client* (RCM 4/8) as well. The RADIUS client and server use the secret to encrypt parts of the packets they send each other, and to guarantee that the messages and replies are authentic. Your RADIUS server might store this list in a text file and in Merit's implementation this is a file called *clients*. 2. A list of authorized users and their configuration information. The RADIUS server needs to know which users have what passwords and what these users are authorized to do after they log in. In Merit's implementation, this is a text file called *users*. Each user is listed along with password (or an indication that the UNIX password file should be consulted), and any restrictions as to which RCM 4/8 or serial ports the user may be allowed to log in from. Information about the user is stored as a list of RADIUS protocol *attributes* and their associated *values*. These translate directly into the authentication reply the server sends back to the client.

RADIUS Accounting

Selecting the ***RADIUS Accounting*** parameter displays the following screen:

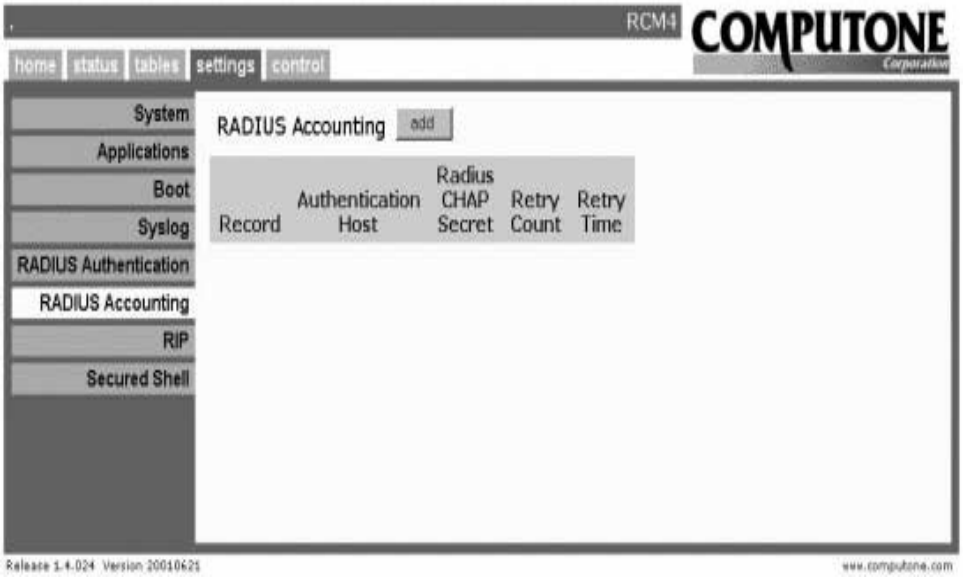


Figure 7 RADIUS Accounting Screen

The following table defines the RADIUS Accounting entries.

Table 8 RADIUS Accounting Screen Entries

Entries	Description
Accounting Host 1	Host name of main RADIUS accounting server or its IP address.
Accounting Host 2	Host name of alternate RADIUS accounting server or its IP address.
Accounting CHAP Secret	Authenticates replies from RADIUS Accounting Server.

RADIUS Accounting occurs when a user logs into or out of a RADIUS *client* after approving the login (either through an internal database or through RADIUS authentication). The client sends notification to the accounting server that this particular user has logged in. When the user logs off or is disconnected, the client also sends notification including the number of seconds the user was connected.

When the RADIUS Accounting server receives these notices, it stores the information and then sends an acknowledgment back to the *client*. If the client does not receive an acknowledgment for its notices, it assumes they were lost and sends out duplicates.

RADIUS authentication can be done without doing accounting, or accounting without authentication. If doing both, the accounting server can be the same host or a different one from the authentication server. Secondary authentication and accounting hosts can also be defined which the RCM 4/8 uses when there is no reply from the primary servers.

RIP

Selecting the ***RIP*** parameter displays the following screen:

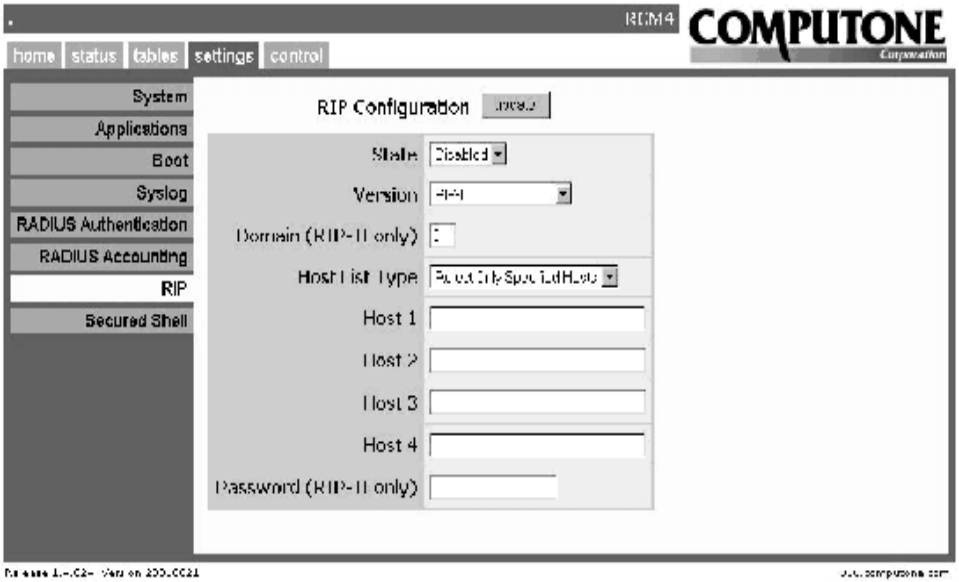


Figure 8 *RIP* Screen

RIP (Routing Information Protocol) is used when the RCM 4/8 needs to share routing information with other hosts. By *listening*, it learns routes from other hosts and by broadcasting or *sending*, it tells other hosts about the routes *it* knows.

The following table defines the entries for the RIP screen.

Table 9 RIP Screen Entries

Entries	Description
State	Enables or disables RIP.
Version	Allows selection of the version of RIP desired.The selections are: RIP - Original version. RIP-II and broadcast - Version II of RIP which sends packets to all hosts on the link. RIP II and multicast -Version II of RIP which sends packets to hosts which have certain IP addresses.
Domain (RIP-II only)	The domain which this server is a member.
Host List Type	Sets the selection process for RIP. The selections are: Reject Only Specified Hosts or Accept Only Specified Hosts.
Host 1	Host 1 to be acted upon, depending on the Host List Type specification. Requires IP address or host name.
Host 2	Host 2 to be acted upon, depending on the Host List Type specification. Requires IP address or host name.
Host 3	Host 3 to be acted upon, depending on the Host List Type specification. Requires IP address or host name.
Host 4	Host 4 to be acted upon, depending on the Host List Type specification. Requires IP address or host name.
Password (RIP-II only)	Password for authenticating RIP II packets.

Secured Shell

Selecting the ***Secure Shell*** parameter displays the following screen:

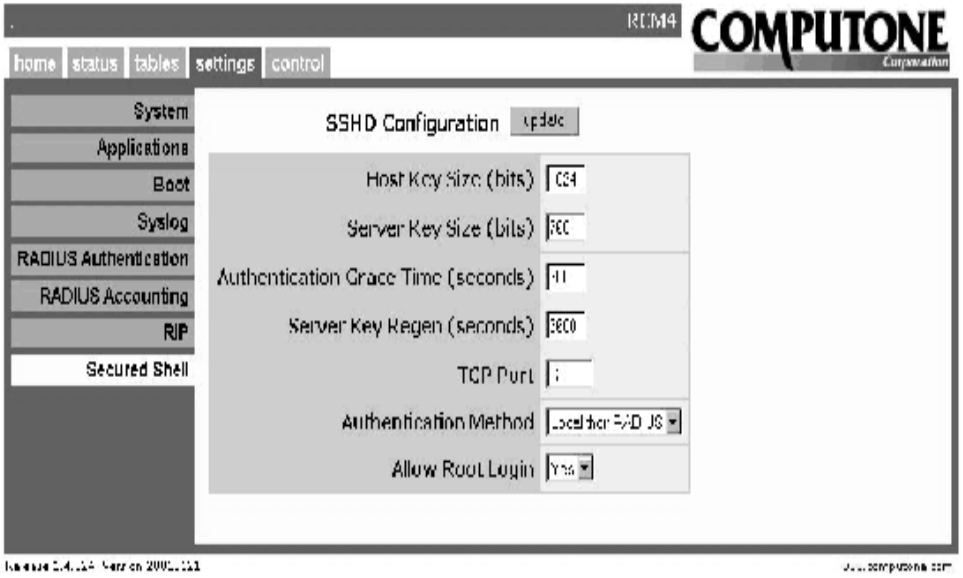


Figure 9 Secured Shell Screen

The RCM 4/8 is shipped configured to have the web server running, as well as telnetd and sshd. However, sshd refuses to run until a host key has been generated, a process that must be started manually. Before generating a host key, however, the secure shell parameters must be configured.

The ***Secured Shell*** screen entries are defined in the following table:

Table 10 Secured Shell Entries

Entries	Description
Host Key Size (bits)	The host key is generated one time and is good for the life of the machine. As such, it gets stored in an area of flash that can neither be saved nor restored.
Server Key Size (bits)	The server key is generated at run time, is stored in memory, and is never used for more than one hour (configurable).
Authentication Grace Time (seconds)	The number of seconds that a user has to successfully authenticate before being cut off.
Server Key Regen (seconds)	Defines the usable life span of a server key in seconds.
TCP Port	Defines the TCP port on which sshd listens for connection requests.
Authentication Method	Defines the method by which users are authenticated. Possible values are **Local**, **Radius**, or **Local then RADIUS**.
Allow Root Login	Either allow or disallow a user to login as root.

Key Size and Security

The host key size and the server key size are paramount in determining how secure the keys are. The following table defines the key type securities.

Table 11 Key Size and Security

Key Type	Strong Security	Weak Security	Notes
Symmetric	128 bits	40 bits	
Asymmetric (Host & Server)	1024 bits	512 bits	It is recommended configuring the host key to be 1024 bits and the server key to be 768 bits. Since the session key is generated by the client, it is suggested that you configure your client (if possible) to generate a 128-bit key. The RCM 4/8 supports up to 2048-bit host and server keys. The practicality of keys larger than that decreased substantially while CPU requirements increased exponentially. Additionally, if you are going to secure a RCM 4/8, be sure to disable the web configuration (httpd) and the insecure shell (telnetd).

The following table defines symmetric and asymmetric key types.

Table 12 Encryption Key Types

Key Type	Definition	Advantage	Disadvantage
Symmetric	When the key used to decrypt the message is the same as the key used to encrypt the message, the key is said to be “symmetric”.	They are fast. Encryption and decryption can be accomplished very quickly when compared to asymmetric keys.	Both the sender and the recipient must agree ahead of time on a specific key.
Asymmetric	Some algorithms derive separate encryption and decryption keys. These are said to be “asymmetric” and are also know as public/private key pairs or RSA keys.	The advantage of asymmetric keys is that you can publish one of them as your “public key” and keep the other one private. Thus someone wanting to send you a message encrypts the message with your public key, knowing that only the person with the private key, namely you, can decrypt it. The reverse can also be used to send a message. If a recipient can decrypt a message using your public key, then the message must have originated and been encrypted by someone with the private key.	The disadvantage of asymmetric keys is that they are extremely math intensive and thus require lots of CPU time.

Configuring Secure Shell Parameters

To begin the secure shell parameter configuration, perform the following steps:

1. Do one of the following: Either telnet into the RCM 4/8, or
   Connect a serial terminal (or terminal emulator) to the console port.
2. From a shell prompt, run the `apps` command to display a list of applications and their current enabled or disabled status.

NOTE: To secure your RCM 4/8, it is suggested that the httpd and telnetd be disabled and the sshd be enabled. To leave your RCM 4/8 unsecured, it is suggested that the sshd be disabled. For the time being, leave httpd and telnetd enabled so that if something goes wrong during the configuration process it can be fixed by telnetting in.

3. Enable the sshd applications with the following command:

```
apps set sshd enable
```

4. Run `ssh` and the ssh parameters are displayed:

Table 13 ssh Parameters

Parameter	Definition
hostkey	Number of bits that are used to generate the host key.
serverkey	Number of bits that are used to generate a server key.
authgrace	Number of seconds that a user has to successfully authenticate before being cut off.
regen	Defines the usable life span of a server key in seconds.
port	Defines the TCP port on which sshd listens for connection requests.
authmethod	Defines the method by which users are authenticated. Possible values are “local”, “radius”, or “both”. In the latter case, the local user file is checked first, then RADIUS.
allowroot	Either allow or disallow a user to login as root.

5. Set the parameters as desired using one or more commands of the form:

```
ssh set <paramName> <value>
```

For more help, type `help ssh`.

6. Save these configuration parameters to flash so that they are used the next time the machine is booted. Saving to a TFTP host is fine as well, as long as the system is configured to fetch its configuration from that same TFTP host. To save, type:

```
save
```

or

```
save <hostname> <filename>
```

All the configuration parameters are now set.

Generating a Host Key

Once the secure shell parameters have been configured, it is time to generate a host key by performing the following steps:

1. Type `sshd gen`

Typical times for a 1024-bit key range from 21 seconds to 4 minutes. The host key is saved to a section of non-volatile memory that is not considered part of the

system's configuration and, thus, is not included in saves and restores. Therefore, it is safe to save or restore your configuration via TFTP.

CAUTION

It is strongly recommended erasing the host key prior to selling the machine, shipping it back for RMA, or otherwise releasing the machine from your possession. An additional argument to `sshd` has been added to help you keep your system secure in the event you should ship it somewhere. This command is:

sshd erase

This erases the host key from non-volatile memory. During the RMA process, Computone can not guarantee the security of your machine, nor the security of your host key. Furthermore, Computone can not guarantee that you'll get the same machine you sent us; it could have a different engine card.

2. Once the host key generation is complete, reboot the machine with the following command:

```
shutdown now
```

This assures that the new configuration takes effect. Depending on the size of your server key, it may take a few minutes after a reboot for sshd to become ready. Recall that it requires both a host key and a server key, and that the server key is generated at runtime.

3. Start your ssh client and connect to the RCM 4/8. Depending on the client it may or may not ask for your user name, though it should always ask for your password. After logging in you should see a shell prompt.

4. To complete securing your RCM 4/8, it is recommended that httpd and telnetd be disabled. This can be accomplished from a shell prompt with the following command:

```
apps set httpd disable
apps set telnetd disable
```

5. Save this configuration to flash and/or a TFTP server:

```
save
```

6. Reboot the RCM 4/8 using the following command to ensure these two services are NOT running:

```
shutdown now
```

CHAPTER 5

Using System Controls

The following table lists the parameters available for configuration through the ***Control*** tab:

Table 1 Control Parameters

Parameter	Description
Shutdown	Allows rebooting the system in the specified number of minutes.
Save to NVRAM	Allows saving of the current working configuration to non-volatile memory.
Save to Host: File	Allows saving of the current working configuration to a TFTP site.
Scan Ports	Allows the scanning of all ports.

Shutdown

Selecting the ***Shutdown*** parameter displays the following screen:

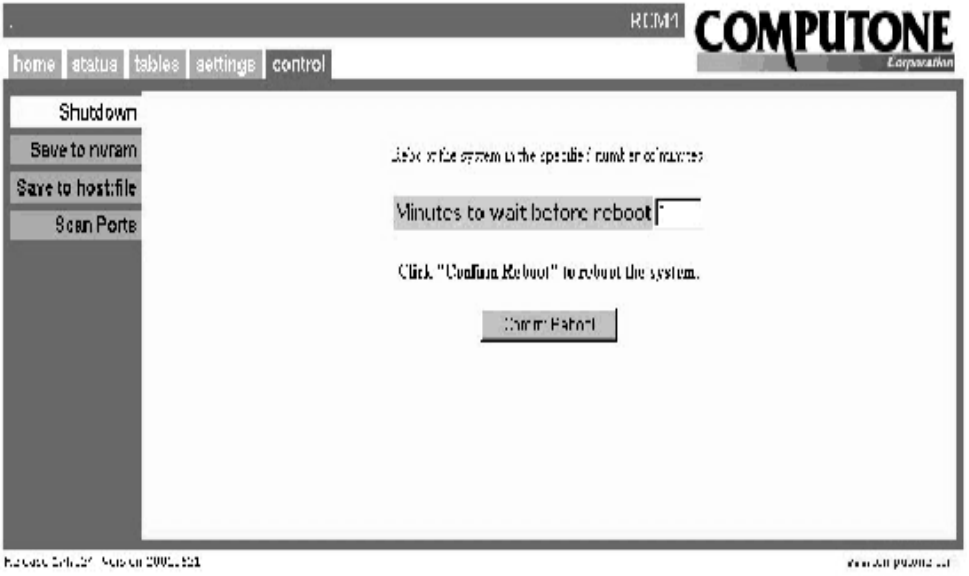


Figure 1 Shutdown Screen

This screen allows the rebooting of the system in the number of minutes that are specified. Click **Confirm Reboot** to reboot the system.

Save to NVRAM

Selecting the ***Save to NVRAM*** parameter displays the following screen:

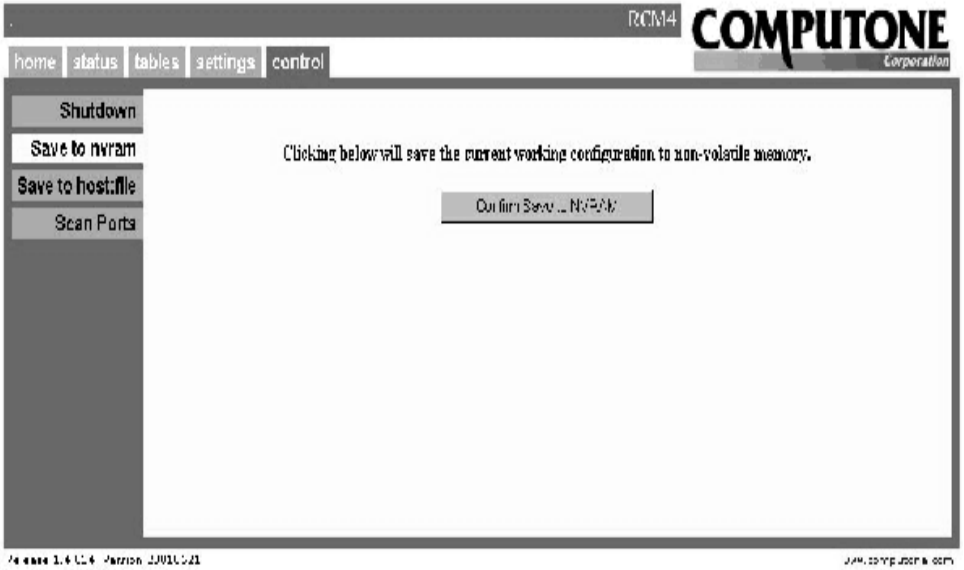


Figure 2 Save to NVRAM Screen

This screen allows the saving of the current working configuration to non-volatile memory (NVRAM). Click ***Confirm save to NVRAM*** to save the current working configuration.

Save to Host: File

Selecting the ***Save to Host: File*** parameter displays the following screen:

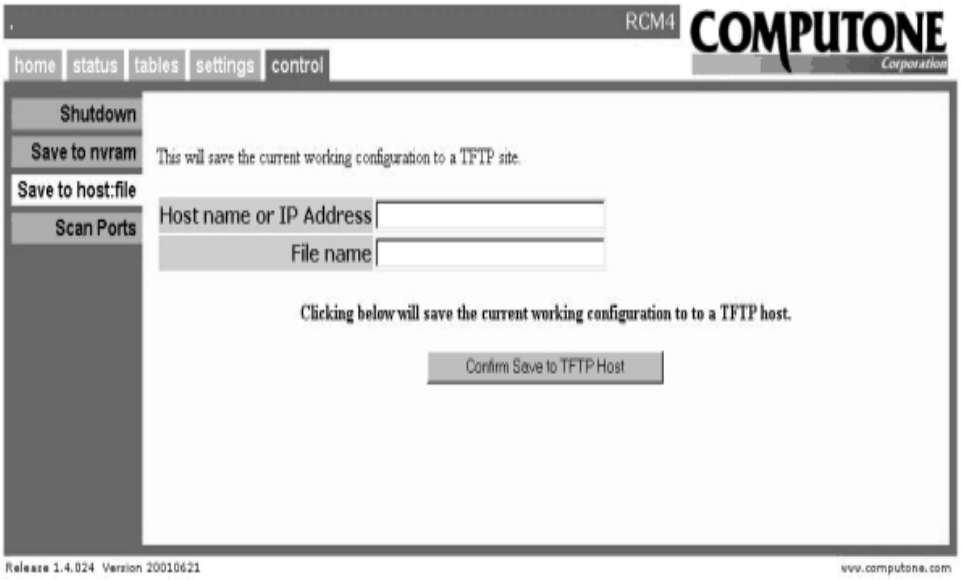


Figure 3 Save to Host: File Screen

This screen allows the saving of the current working configuration to a TFTP site. The host name or IP address must be entered where the file is to be stored along with the desired file name.

Scan Ports

Selecting the ***Scan Ports*** parameter displays the following screen:

CHAPTER 6

Command-line Interface

This section shows the commands available at the command-line interface and the command usages.

The Commands

Table 1 shows the list of commands that can be issued at the command line.

Table 1: IntelliServer Commands

Command	Description
apps	Provides ability to enable/disable httpd, sshd, telnetd, and rcpd.
arp	Modify or display the arp table.
clear	Clear the screen.
exit	Log out.
help	Displays the help screen.
killport	Send a kill signal to all processes running on the selected ports. This causes the processes to stop immediately, without any opportunity for clean-up.
logout	Log out.
netstat	Display network statistics.
password	Change password.
ping	Check connection to remote host.
ramstat	Display RAM statistics.
restore	Restore the IntelliServer's configuration.
route	Modify or display dynamic routes to other hosts or networks.
save	Save the IntelliServer's configuration.
scanports	Send a signal to all port managers to rescan and act on the port current configuration.
shutdown	Initiate shutdown and reboot.
telnet	Initiate a telnet session to a host.
tip	Connect to a serial port.
tty	Shows the port accessed.
version	Display IntelliServer version.
whodo	Display session information.

Usage

This section explains how to type each command to access the information or procedure desired.

apps

The ***apps*** command is used to enable or disable httpd, sshed, telnetd or rcpd**.**

Command:	apps
Usage:	# show apps # set apps
Type:	# show apps
Result:	set apps httpd Enabled set apps sshd Disabled set apps telnetd Enabled set apps rcpd Enabled

The following is an example of changing the state of one of the applications (sshd):

Type:	# set apps sshd enabled
Result:	Use the **show apps** command to verify. The following is displayed: set apps httpd Enabled set apps sshd Disabled set apps telnetd Enabled set apps rcpd Enabled

arp

To add or delete ARP table entries by hand, use the add ***arp*** and the delete ***arp*** commands, accordingly.

Command:	**arp**
Usage:	# arp show [<ip address>|<hostname>|-n] # arp add <ip address>|<hostname> <etheraddr> [temporary] [published] [trailers] # arp delete <ip address>|<hostname # arp flush [all]
Type:	# arp show
Result:	cton.computone.com (160.77.1.10) at 00:10:5a:68:87:a4, 0 minutes billl.computone.com (160.77.24.208) at 00:10:5a:68:87:df, 0 minutes

Entries can be added manually using the ***add arp*** command. If adding *permanent* entries, they will not be removed unless they are deleted manually with the ***delete arp*** command. This command can also be used to manually delete entries added by other means.

Permanent ARP entries are only permanent as long as the IntelliServer stays up. They are not stored in NVRAM the way static routes in the *Gateway Table* are, for example. This is generally not an issue, because usually the only permanent entries dealt with are the ones created automatically for proxy ARP on behalf of remote hosts. *Temporary* means **not** permanent. That is, if using the ***add arp*** command, it is assumed permanent unless the temporary option is added. This is because entries added by hand are permanent until you delete them.

Type:	# arp add 160.77.99.4 00:00:c0:7e:ee:30 published
Result:	To verify, type: # arp show (160.77.99.4) at 00:00:c0:7e:ee:30, 0 minutes, permanent, published cton.computone.com (160.77.1.10) at 00:10:5a:68:87:a4, 0 minutes billl.computone.com (160.77.24.208) at 00:10:5a:68:87:df, 0 minutes

Type:	**# delete arp 160.77.99.4** **# delete arp jeeves**
Result:	The arp entry known by jeeves, or IP address 160.77.99.4, is deleted.

To remove all arp entries, use the following command.

Type:	**# arp flush all**
Result:	The entries in the arp table are all deleted.

clear

The ***clear*** command erases the screen.

Command:	**clear**
Usage:	clear
Type:	# clear
Result:	Clears the screen.

exit

The ***exit*** command closes the telnet session to IntelliServer.

Command:	**exit**
Usage:	exit
Type:	# exit
Result:	Closes telnet session to IntelliServer.

help

The ***help*** command displays the **help** screen.

Command:	**help**
Usage:	help <command>
Type:	# help
Result:	apps - Provides ability to enable/disable httpd, sshd, telnetd, and rcpd arp - Modify or display the arp table exit - Logout help - Display this help screen killport - Sends a kill signal to all processes running on the selected ports. This causes the processes to stop immediately, without any opportunity for clean-up. logout - Logout netstat - Display network statistics password - Change password ping - Check connection to remote host ramstat - Display RAM statistics restore - Restore the IntelliServer's configuration route - Modify/Display dynamic routes to other hosts or networks save - Save the IntelliServer's configuration scanports - Sends a signal to all port managers to rescan and act on the port current configuration. shutdown - Initiate shutdown and reboot telnet - Initiate a telnet session to a host tip - Connect to a serial port tty - Shows what port you are on version - Display IntelliServer version whodo - Display session information #

killport

The ***killport*** command sends a kill signal to all processes running on the selected ports. This causes the processes to stop immediately, without any opportunity for clean-up.

Command:	killport
Usage:	# killport 1 # killport 2-4 # killport pts01 # killport all
Type:	# killport 1 # killport 2-4 # killport pts01 # killport all
Result:	line 1 - port number 1 processes are killed line 2 - ports 2-4 processes are killed line 3 - port named pts01 processes are killed line 4 - all processes on all ports are killed

logout

The ***logout*** command terminates the telnet session to the IntelliServer.

Command:	logout
Usage:	logout
Type:	# logout
Result:	Terminates telnet session to IntelliServer.

netstat

Network protocols are designed with the knowledge that network traffic is not always perfect. Packets may be damaged or delayed in transmission, data may be incorrectly routed, and host machines can be misconfigured. The different modules that comprise the IntelliServer's networking software keep records called *network statistics*. They count the number of packets which cross into their territory, noting especially those that are not proper, or which present any difficulties. Then, when the system administrator notices a problem with the network, these statistics can be reviewed by using the ***netstat*** command. By seeing which modules report difficulties and which do not, clues sometimes become available as to where to look for the problem.

A single protocol, a combination of several, or all the protocols can be reviewed. The following is an example of requesting statistics for a single protocol and then for two protocols.

Command:	netstat
Usage:	netstat tcp|ip|icmp|udp|route|sonic|ppp|slip|connections|all
Type:	# netstat udp
Result:	-- udp -- 43483 input packets 22 output packets 0 bad header pullup 0 bad checksum 0 bad length
Type:	netstat tcp ip
Result:	-- udp -- 43483 input packets 22 output packets 0 bad header pullup 0 bad checksum 0 bad length -- ip -- 48092 total pkts rcvd 1792 total pkts sent 0 bad checksum 0 cant pullup hdr 0 less data than hdr says 0 bad hdr len 0 data len less than a hdr 0 frags rcvd 0 frags dropped 0 frags timed out 0 pkts forwarded 57 cant forward to dest 0 redirects frwrd same net 0 pkts lost

password

When the ***password*** command is used, prompting for the password occurs. If it won't echo back, prompting occurs again for confirmation.

Command:	password
Usage:	password <name> alias of "user set <name> password <passwd>" with prompting Set password to NOPROMPT to omit password prompt
Type:	# password Changing password for root on New password: Re-enter new password:
Result:	The login password is changed.

To Omit the Password Prompt

Normally, the IntelliServer prompts for a password during login, even if no password was configured (in which case the proper response is to press enter without typing a password). To configure a user so when that user name is given, there is no password prompt, configure an NVRAM user to skip the password prompt entirely. Set the password to **NOPROMPT** (exactly as shown - all upper-case - all one word).

ping

The ***ping*** command sends an *ICMP Echo Request* packet to the designated host and waits for a reply. When it receives the reply it reports that the host is *alive*. This is a basic command to use when you want to verify that two hosts can communicate with each other.

Command:	**ping**
Usage:	ping [-s|t] <hostname>|<ip address> -s pings each second & reports statistics -t traces route
Type:	# ping cton
Result:	cton.computone.com (160.77.1.10) is alive

A host is pinged using its host name (**ping cton**) so the IntelliServer had to resolve the name into an IP address. This it did either through its own Host Table or by sending a name resolution request to one of the name servers. Once the IP address was determined, the echo request was sent and a reply received. The IntelliServer shows both the host name you supplied (**cton**) and the IP address it found (**160.77.1.10**). If the IntelliServer can not find the host in its local table, nor obtain the IP address from a name server, it replies *<host name> unknown host*.

port

The ***port*** command is very versatile and is used to modify or display configuration of a port.

Command:	port
Usage:	show port [<port-list> [full|access|hardware|options|counts]] set port <port-list> from <number> set port <port-list> {parameter <value>} {parameter <value>} options are: [login byport|byscreen|auto|autowait|printer|revtcp|outbound|byporttcp] [comment <comment>] [term <type>] [username <username>] [rterm <type>] [group <group#>|none] [autoppp enabled|disabled] [modem enabled|disabled] [wait enabled|disabled] [init <initstring>] [speed <speed>] [charsize 5|6|7|8] [dialscript <scriptname>] [parity none|even|odd|space|mark] [stopbits 1|1.5|2] [inflow disabled|xoff|rts|xoffrts] [outflow disabled|xon|xany|cts|xoncts|xanyxts] [oxlat disabled|nl_crnl|cr_nl|strip_cr|crnl_crnl] [ixlat disabled|cr_nl|nl_cr] [tabs enabled|disabled] [intr <char>] [erase <char>] [kill <char>] [tcp normal|crnl_cr|raw] [iview <profile>] [iset <profile>] [iprint <profile>]

Command:	show port
Usage:	show port [<port-list> [full|access|hardware|options|counts]]
Type:	# show port 1
Result:	Displays the current requested values of the specified serial port(s). The following is displayed: ! ! Ports ! Key: 1 ! set port 1 comment "" set port 1 type Disabled set port 1 user root set port 1 group None set port 1 localterm unknown set port 1 remterm "" set port 1 modem No set port 1 waitinput Yes set port 1 dialup "" set port 1 modeminit "" set port 1 speed 9600 set port 1 databits 8 set port 1 parity None set port 1 stopbits 1 set port 1 autoppp No set port 1 inflow None set port 1 inxlat "CR to NL" set port 1 outflow None set port 1 outxlat "NL to CR+NL" set port 1 tabs No set port 1 tcpmode Normal set port 1 intrchar ^C set port 1 erasechar ^H set port 1 quitchar "^\\" set port 1 killchar ^U set port 1 eofchar ^D set port 1 viewprof "" set port 1 printprof "" set port 1 setprof ""

Command:	set port
Usage:	set port <port-list> from <number>
Type:	# set port 1 from 3
Result:	Copies the port settings from port 3 to port 1. The following is displayed: Changes will take place at next login

Command:	set port
Usage:	port set <port-list> {parameter <value>}
You Type:	# port set 1-7 {parameter <value>} {parameter <value>} {parameter <value>}…
Result:	Because of the many combinations of login types, term types, modem, user names, physical port characteristics, input flow control options, output flow control options, output processing, input processing, special keys, reverse TCP options, and IntlliFeature selections, it is near impossible to list the total combinations possible.

NOTE: If setting the login type, use the following syntax:

Usage:	set port <port-list> {parameter <value>}
You Type:	# port set 1 type \”auto login\”

The following table defines the {parameter <value>} pairs shown above.

Table 2: Port Set {parameter <value>} Pairs

Login Types:	Results
disabled	Nothing happens on this port, except that commands can be used such as **tip** and **output** to send data to it in order to test the port or configure a modem or other device.
login <byport>	With this selection, the port sends a login prompt to the attached terminal or modem. When the user logs in, the IntelliServer starts up whatever connections have been configured for that user.
login <byscreen>	Generally use this setting only if the port is configured to support multiple sessions through IntelliView. Each virtual screen is sent its own login prompt, and the user must log in to each virtual screen separately. When a session is ended, a new login prompt for that virtual screen is sent, but DTR is not dropped if there is any other session on this port still active.
login <autowait>	This is almost identical to *Auto-Login*, except that instead of launching the connection immediately, the port first sends a prompt: **Press <enter> to continue**, and when the operator does this, *then* the connection is launched. This is designed to solve the quandary that occurs when a port configured as *Auto-Login* is attached to a local terminal that is always on, but frequently unattended. The user logs off and walks away, and the IntelliServer immediately launches the connection. Suppose that connection is an attempt to rlogin to some host machine. So that machine prompts for a password. Since there is no one present to enter a password, the connection soon times out and is restarted, and times out and is restarted and so on. If the port is configured as *Auto-Login, wait*, then the IntelliServer remains at the "**Press <enter> to continue**" prompt until someone does this and the retries and time outs are avoided.
login <auto>	To configure a port so that the IntelliServer automatically starts a connection without prompting for a login, perform the following procedures. If the port is configured for *Auto-Login*, specify a user name for this port. The port behaves exactly as a *Login-by-Port* but instead of sending a login prompt, assumes that the specified user has successfully logged in and starts up connections accordingly. When the session is over, the IntelliServer will (after waiting for *carrier* when appropriate) restart the sessions again.
login <printer>	This configuration is similar to *Reverse-TCP*, except that a port configured as a *printer* can accept connections from **rcp** and **rsh cat** clients on your network.

Table 2: Port Set {parameter <value>} Pairs

login <revtcp>	When a port is configured as *Reverse-TCP*, the IntelliServer accepts a TCP connection from some other host on the network. Data received from that host is sent out the serial port and data received from the serial port is sent to the host. This is a common method of supporting printers and other "non-login" serial devices.
login <outbound>	This configuration supports outbound PPP/SLIP/CSLIP links. The IntelliServer brings these links up automatically when it tries to route a packet to a network that it knows to be on the other side of one of these links. Note that this port type supports only *dial-out* connections. To support clients that are dialing into the IntelliServer, configure the port as *Login-by-Port*.
login <byporttcp>	This is a combination of *Login-by-Port* and *Reverse-TCP* and is designed to support bidirectional operation of a modem. Configure the port as *modem enabled*, because the IntelliServer uses *carrier* (DCD) to sense incoming calls and determine whether there has been a disconnection. When the port is idle and there is no incoming call, the IntelliServer accepts TCP connections for this port from hosts on the net, just like *Reverse-TCP*. If a connection is established, the client can access the modem, send dialing commands, and connect to other systems. Anyone trying to dial in gets a busy signal because the modem is off-hook. If an incoming call comes in first, the port sends out a login prompt, like *Login-by-Port*, and as long as the incoming call is connected the IntelliServer refuses or defers TCP connections from the network for that port.
comment <comment>	Your comment to identify this port.
term <type>	This defines the default terminal name that is sent when you telnet or rlogin from this port to a host on your network. This default value may be overridden by other settings, however. Because this information is used by the menus, the IntelliServer needs to understand the terminal characteristics that each terminal name represents. There are a limited number of terminal types supported. Your choices are: unknown wyse30 xterm ansi wyse50 uterm0 uterm1 vt100 wyse60 uterm2 uterm3 The last four terminal types are user-definable. If your terminal does not emulate one of the defined terminals, store your terminal's definitions under one of these four terminal types.
user name <user name>	Name of port user.

Table 2: Port Set {parameter <value>} Pairs

rterm <type>	If you enter a name here, then by default it is sent when you rlogin or telnet to a host, instead of using the one given for the *Local terminal name*. Since the RCM 4/8 does not need to know what this name actually means, it can be any name that is understood by the login host. The telnet and rlogin commands also support command-line arguments which, if used, can override these default terminal-types. If there is no command-line argument, the *remote term type* is used, and if no *remote term type* is defined, then the *local term type* is sent.
autoppp enabled|disabled	When a port is set for login, it normally waits for a user name then prompts for a password. This information is used to determine who is logging in and what service should be invoked (shell, ppp, telnet, etc.). If the user is defined as a ppp user, then a ppp process is started and the username and other information is passed to the ppp process. Some systems don't go through this "normal" login process when running ppp. They simply start sending ppp packets and the ppp protocol sorts out the user name and other information. Autoppp is designed to handle this case. While login is looking for a user name, it also looks for the start of a ppp packet if the autoppp flag is set. Upon receiving a ppp packet, login automatically invokes ppp to handle the link.
Modem	
modem enabled|disabled	Select **modem enabled** for dial-in connections. The IntelliServer waits for the modem to assert **DCD** before it sends the login prompt. For dial-in and dial-out connections, if **DCD** is dropped the IntelliServer recognizes that there has been a disconnection and terminates whatever processes have been using that port. For login ports, the user is marked as having logged off, for reverse-TCP ports the TCP connection is closed. When a port is configured in this way, it is called a “modem port”, even though it may be connected to a terminal or other equipment. The default is **modem disabled** and this means that the IntelliServer is to pretend that the **DCD** signal is always asserted. This is intended for connections to local terminals, printers, etc., and is sometimes called a “non-modem” port.
wait enabled|disabled	Normally a port configured to send a login prompt will do so shortly after detecting that **DCD** is asserted. If you set this to **enabled**, then after detecting carrier the IntelliServer waits until it receives some incoming data on the port before it sends the login prompt. There are some modems which raise **DCD** while they are still in command mode, which prevents the modem from mistaking the login message for a modem command. This is in case the built-in delay between sensing carrier and sending the prompt is not long enough.

Table 2: Port Set {parameter <value>} Pairs

init <initstring>	This setting is used by ports that are configured for terminals or dial-in connections. It defines a string of commands which the IntelliServer transmits to the modem before it waits for the next incoming call and is not always required. Some modems can be configured ahead of time and never seem to lose their settings. When does the string get sent? A user logs off, the IntelliServer drops **DTR** to hang up the line, waits a second, raises **DTR,** sends the initialization string and then waits for modem to assert **DCD**. A call comes in, next user logs in, works, logs off and so on.
Physical Port Characteristics	
speed <speed>	This sets the line speed at which data is transmitted and received; *speed* must be one of the following: 50 150 1200 3600 19200 64000 75 200 1800 4800 38400 76800 110 300 2000 7200 56000 115200 134.5 600 2400 9600 57600 In addition, define custom rates by setting up an IntelliSet profile and assigning that profile to a port. By using IntelliSet, the ability to specify a *split baud-rate*, where the port transmits at one speed and receives at another is available. When line speeds and other parameters are defined using IntelliSet, those values over-ride the ones chosen here.
charsize 5/6/7/8	This sets the number of data bits per character. Size must be 5, 6, 7, or 8 bits.
dialscript <scriptname>	This is the name of a dialer script which is used by ports configured for outbound PPP/SLIP/CSLIP links. It specifies the commands that have to be sent to the modem so it will dial and establish a connection, and, allows the IntelliServer to wait for particular responses. Different modems may require different dialer scripts. That is why the dialer script is stored on a per-port basis, while the *login script* (which depends on the particular target of the call) is identified in the *remote profile*.
parity none|even|odd|space|mark	This controls the parity bit sent with each characters. Parity must be one of the following: none, even, odd, space, or mark.
stop bits 1|1.5|2	This controls the number of stop bits that are transmitted after each character. Choices are **1**, **1.5**, or **2** bits. One stop bit is generally sufficient except when you are connecting to devices that are very old, very slow, or very unusual. This has no effect on the receiver, since one stop bit is always sufficient.
Input Flow Control Option	
inflow disabled	There is no input flow control. No attempt is made to notify the sender when the receive-buffers are becoming full. This is often used with terminals because key-strokes do not usually arrive fast enough to overrun computers.

Table 2: Port Set {parameter <value>} Pairs

inflow xoff	When the port's receive buffer becomes mostly full, an XOFF character is sent to the sending device to tell it to stop sending data. When the buffers start to get empty again, an XON character is sent to tell it to restart transmission.
inflow rts	When there is room in the IntelliServer receiver buffer, the *request-to-send* **(RTS)** data-set signal is asserted. When the buffers become mostly full, this signal is negated ("dropped"). When connecting to a modem, wire the IntelliServer RTS output to the modem's RTS input, and configure the modem for RTS flow control. (You are probably using CTS flow control as well.) When connecting to a device, like another IntelliServer port configured for *CTS outflow,*connect the IntelliServer RTS to the devices CTS (and probably vice-versa).
inflow xoffrts	***Inflow*** controls the flow of data into our port. A setting of xoff means that the IntelliServer sends an xoff character to the other side of the link to signal the device attached to our port to stop sending data until further notice. RTS means that the IntelliServer lowers, or deactivate, the RTS signal to stop the flow of data coming into the port. ***Xoffrts*** means the IntelliServer does both to stop the flow of data into the port.
Output Flow Control Options	
outflow disabled	Output flow control is disabled. The IntelliServer does not recognize any condition that means "stop sending."
outflow xon	If an *XOFF* character is received (normally a byte with the binary value 10011), the IntelliServer stops sending data until an *XON* character is received (binary 10001). These flow control characters are also stripped from the data stream and are never seen as actual "incoming data". This is sometimes called XON/XOFF flow control, sometimes "software flow control" because it was traditionally implemented in software. It is also sometimes called "in-band" flow control, because the flow-control information is sent along the same wires as the data itself. This form of flow control can be used with terminals and printers and some types of file transfer, when the normal data passing between the devices does not contain bytes equal to the XON and XOFF values. This type of flow control is usually not suitable for PPP/SLIP connections, or binary file transfers that would be carrying data which might contain XON/XOFF bytes.
outflow xany	This is a variation on XON/XOFF flow control intended for terminals. As with the previous, receiving an XOFF character makes the IntelliServer stop transmitting. Once the IntelliServer has stopped, the receipt of any data (including a second XOFF) causes the IntelliServer to resume. Since the default XON and XOFF values correspond to the **ctrl-Q** and **ctrl-S** on traditional ASCII keyboards, this corresponds to the MSDOS convention of entering **ctrl-S** once to suspend output and again to resume.

Table 2: Port Set {parameter <value>} Pairs

outflow cts	Data is sent as long as the *clear to send* **(CTS)** data-set signal is asserted (raised). When **CTS** is negated (dropped) the IntelliServer stops sending data until the signal is asserted again. This is sometimes called "hardware flow control" because it was traditionally implemented in hardware. It is also called "out-of-band" flow control because the flow control information is sent on a separate wire from the data. This type of output flow control is recommended for PPP/SLIP/CSLIP links or any connections where arbitrary binary data that could include XON or XOFF characters are being transferred.
outflow xoncts outflow xanycts	Combinations of XON and CTS, or of XANY and CTS. The IntelliServer does not transmit data unless *both* conditions permit. A good example where this might be used is for a slower terminal (which needs robust flow control) whose operator wants to use **ctrl-S** to suspend and resume output. Many terminals have scroll-lock keys, however, and in that case CTS flow control alone would suffice.
Output Processing	
oxlat disabled	With this setting, the port will send carriage return and linefeed characters as-is with no translation (ascii CR and LF, respectively).
oxlat nl_crnl	With this setting, a carriage-return (**CR**) is added after any linefeed in the output (also called a "newline"). This is useful when printing output from systems like UNIX, which use a single linefeed character to delineate the ends of lines. If you send such output directly to most printers, each new line begins directly below where the previous line left off, creating a "barber-pole" effect.
oxlat cr_nl	Carriage return characters are changed to linefeeds (newlines).
oxlat strip_cr	Carriage returns which occur at the beginning of a line are thrown away.
oxlat crnl_crnl	Carriage returns are added before any newline, and newlines are added after any carriage return. In other words, *either* a carriage return *or* a newline becomes *both* a carriage return and a newline.
tabs enabled|disabled	With this setting, the port translates ascii tab characters to a sequence of spaces sufficient to achieve tab stops at 8-character intervals. This tab setting corresponds to the traditional tab processing performed on UNIX systems and is useful when printing output from a UNIX system using tools that expect this processing to be performed "downstream". If this parameter is set to **disabled**, then tab characters are sent unchanged.
Input Processing	
ixlat disabled	Using this option, no input processing is performed.
ixlat cr_nl	The carriage return key (ascii CR) is mapped to a linefeed character (ascii LF). This is the default and the only reason I can think of to change it would be to sup-port a terminal that sends both carriage-return and linefeed when the Return key is pressed.
ixlat nl_cr	We are still trying to decide what this one is for.

Table 2: Port Set {parameter <value>} Pairs

Special Keys	
intr <char>	This defines the *interrupt key.* Use this key to quickly terminate commands before they have finished. In this example **^c** represents *control-c*. In either the menu or the command line, you type ^ and **c** to enter the value as shown. You could also enter *control-c* itself, unless it were already defined as one of the special keys. The **del** key has a special representation.
erase <char>	This defines the character used to backspace a single character and erase it.
kill <char>	This defines the character used to cancel the entire line you are currently typing. This is used when you are doing line-oriented input such as at the command prompt or in telnet command mode.
Reverse TCP Options	
tcp normal|crnl_cr|raw	Normally, a reverse-TCP connection uses telnet protocol. Telnet server implementations differ in their treatment of carriage-return (CR) and linefeed (or new-line, NL) characters. With some, if a CR-NL pair is received from the network, the two characters will be output. That is what the *normal* option does. With other telnet servers, if a CR-NL pair is received, the CR is sent but the NL is ignored. This is the *CRNL->CR* option. These two options are provided for maximum compatibility. The third option, *Raw,* causes the Reverse-TCP connection on that port to not use telnet protocol at all. Instead, the data received over the TCP connection is sent to the port exactly as received, and vice-versa. This is provided for compatibility with other vendors' products, as well as providing an easy-to-use interface for special applications.
group <group#>|none	There are 16 groups of ports, numbered 0 to 15. Any port can belong to any group or to no group at all. When something tries to start a reverse-TCP connection to the IntelliServer, it can specify a particular port or a particular port group. When a port group is specified, the first available port in the group is used. A port group number can also be specified in a *Remote Profile* for an outbound PPP/SLIP/CSLIP interface. A port is configured as *Reverse-TCP* or *Login-by-Port/TCP* cannot be a member of the same group as a port configured as *Printer* or which uses *IntelliPrint*. This is because the first types suppress output processing, while the others perform it. If both types were members of the same group, the results might depend on which printer happened to be available.
IntelliFeatures	
iview <profile>	This specifies the name of the IntelliView profile you want to apply to this port.
iset <profile>	This specifies the name of the IntelliSet profile desired to apply to this port.
iprint <profile>	This specifies the name of the IntelliPrint profile desired to apply to this port. It can be any print profile that has already been created.

Command:	port output
Usage:	port output [port] string <text>[forever]
Type:	# port output 1 string “Hey, did you get this message?”
Result:	The message “Hey, did you get this message?” is output to port 1 one time. If the [forever] option is used, the message is output continuously and locks-up the port.

Command:	port output
Usage:	port output [port] pattern barber|columns [forever]
Type:	# port output 1 pattern barber
Result:	Sends barber pole pattern to a specified port. The following is output: !@#$%^&*()-1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[]. @#$%^&*()-1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].! #$%^&*()-1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@ $%^&*()-1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@# %^&*()-1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$ ^&*()-1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$% &*()-1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^ *()-1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^& ()-1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*)-1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*(-1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*() 1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*()- 234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*()-1 34567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*()-12 4567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*()-123 567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*()-1234 67890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*()-12345 7890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*()-123456 890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*()-1234567 90ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*()-12345678 0ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*()-123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*()-1234567890 BCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}/[].!@#$%^&*()-1234567890A

RAMSTAT

The ***RAMSTAT*** command displays the RAM statistics.

Command:	**ramstat**
Usage:	ramstat
You Type:	# ramstat
Result:	Streams Resources: Resource Name / Number Available / Number In Use / Total Allocated / Times Used / Fail / Memory Source Streams 0 28 28 40 0 Streams Queue Pairs 1 70 71 95 0 Queue Pairs Links 0 14 14 14 0 Generic 24 Events 0 0 0 0 0 Generic 16 Message Blocks 23 153 176 13642 0 Message Blocks Size 16: 23 2 25 2552 0 Generic 32 Size 32: 3 4 7 1963 0 Generic 48 Size 64: 5 1 6 4183 0 Generic 80 Size 152: 0 0 0 0 0 Generic 168 Size 256: 5 18 23 26 0 Generic 272 Size 560: 3 128 131 4910 0 Generic 576 Size 1344: 0 0 0 0 0 Generic 1360 Size 2024: 0 0 0 0 0 Generic 2040 number of bufcalls: 0 total queues 140

Result: (Cont.)

PID	TEXT Ref/Size(K)	DATA Type/Size(K)	HEAP Flags/Size(K)	DRAM Total(K)	Name
1	1 8	S1 4	---- 0	8	init
2	1 8	S1 4	0020 8	16	netmgr
603	1 4	S1 4	---- 0	8	syslogc
4	1 8	S1 4	---- 0	8	telnetd
5	1 12	S1 4	0020 12	20	logger
6	2 40	S3 12	0020 8	24	ksh
7	1 4	S1 4	---- 0	8	rshd
8	2 8	L 16	---- 0	20	ttyd
9	2 8	L 16	---- 0	20	ttyd
610	2 40	S3 12	---- 0	16	ksh
11	1 12	L 16	0020 8	28	portd
7212	1 8	L 24	--- 0	28	ramstat
13	1 4	S1 4	0020 8	16	sessmgr
14	1 4	S2 8	---- 0	12	rspd
15	1 16	L 24	---- 0	8	httpd

Plus DRAM for Text Segments = 64 K
Total DRAM allocated to processes = 324 K

Result: (Cont.)

Kernel Memory Resources:

Resource Name	Free	In Use	Size	Flags	Memory Allocation
Generic 16	0	0	16	.S..	0 (0 x 4K)
Generic 24	120	50	24	.S..	4096 (1 x 4K)
Message Blocks	162	178	24	U...	8192 (2 x 4K)
Control	0	0	32	U..Z	0 (0 x 4K)
File Pointers	83	171	32	U...	8192 (2 x 4K)
Generic 32	102	25	32	.S..	4096 (1 x 4K)
Generic 48	78	7	48	.S..	4096 (1 x 4K)
Protocol Blocks	56	7	64	U..Z	4096 (1 x 4K)
Generic 80	45	6	80	.S..	4096 (1 x 4K)
Streams	23	28	80	U...	4096 (1 x 4K)
Queue Pairs	13	71	96	U...	8192 (2 x 4K)
TTY	0	0	104	U..Z	0 (0 x 4K)
VSM	32	4	112	U..Z	4096 (1 x 4K)
Sockets	29	5	120	U...	4096 (1 x 4K)
User	25	1	152	U..Z	4096 (1 x 4K)
TCP	20	5	160	U..Z	4096 (1 x 4K)
Generic 168	21	3	168	.S..	4096 (1 x 4K)
Generic 272	7	23	272	.S..	8192 (2 x 4K)
Generic 576	2	131	576	.S..	77824 (19 x 4K)
Generic 1360	0	0	1360	.S..	0 (0 x 4K)
Generic 2040	0	0	2040	.S..	0 (0 x 4K)

Total DRAM allocated for dynamic O/S: 152 K

DRAM Allocation Pool Usage			Overall DRAM Usage		
Free Space	= 5380 K	91.4%	DRAM Allocation Pool	= 5884 K	71.8%
Kernel Usage	= 152 K	2.5%	Kernel Fixed Allocation	= 1118 K	13.6%
Process Usage	= 324 K	5.5%	Kernel Static DRAM Usage	= 116 K	1.4%
			Binary Load Image	= 946 K	11.5%
Other	= 28 K	0.4%	Bootloader	= 128 K	1.5%
Total Size	= 5884 K	100.0%	Total DRAM Installed	= 8192 K	100.0%

restore

The ***restore*** command restores the IntelliServer's configuration.

Command:	restore
Usage:	+restore +restore <hostname> <filename> +restore factory
Type:	# **restore** # restore 160.77.99.3 /usr/lib/isconf. # restore factory
Result:	line 1 - restore local configuration. line 2 - restore configuration from a remote host. line 3 - restore factory defaults.

route

The ***route*** command allows adding and deleting the routing table entries manually.

Command:	route
Usage:	+route add <ip address>|<hostname>|<network>|default [address] <ip address>|<hostname> +route delete <ip address>|<hostname>|<network>|default [address] <ip address>|<hostname> route show
Type:	# route add 160.88.128.0/17 160.77.99.35 # route delete 160.88.128.0/17 160.77.99.35 # show route
Result:	1 st line - route is added from network address to host ip address 2 nd line - route is deleted from network address to host ip address 3 rd line - route is displayed
+	Must have administrative privileges to use this command.

Routes need to be manipulated manually, because **s**ometimes there is a need to correct a configuration error made in the gateway table or RADIUS routes.

Rather than fix the gateway table or RADIUS user file and then have to bring up the link again, it may be possible to patch the routing table on the quick, to make sure the change has the desired effect.

Changes to the routing table are immediately effective. This procedure can also be done to adapt to changes in your network configuration.

save

The ***save*** command stores the IntelliServer's configuration.

Command:	**save**
Usage:	save +save <hostname> <filename>
Type:	# **save** **# save 160.77.99.3 /usr/lib/isconf.3** **# save jeeves /tmp/j34**
Result:	1 st line - save configuration locally 2 nd line - save host ip address filename 3 rd line - save hostname filename
+	Must have administrative privileges to use this command.

scanports

The ***scanports*** command sends a signal to all port managers to rescan and act on the port current configuration.

Command:	**scanports**
Usage:	scanports
Type:	# **scanports**
Result:	A signal is sent to all port managers to rescan and act on the current port configuration.

shutdown

The ***shutdown*** command is port related. After sending warnings to every active port (omitting printer ports and outbound SLIP/PPP connections) it then kills all the processes and shuts down the IntelliServer.

Command:	shutdown
Usage:	+shutdown now|<time> [<message>]
Type:	# **shutdown now "Shutting down NOW!"** # **shutdown 1 "Shutting down in 1 minute."**
Result:	1 st line - causes a shutdown and reboot now and sends message shown. 2 nd line - causes a shutdown and reboot in 1 minute and sends message shown.
+	Must have administrative privileges to use this command.

If configuration changes have been made and not yet saved in NVRAM, a warning appears and asked for confirmation before the shutdown proceeds. As the shutdown time approaches, repeated warnings are sent to any users still logged on. These warnings include the optional message specified with the shutdown command. Any ports still active at zero hour are killed.

sshd

The ***sshd*** command is the secure shell host key generator and daemon. The RCM 4/8 is shipped configured to have the web server running, as well as telnetd and sshd. However, sshd refuses to run until a host key has been generated, a process that must be started manually. Before generating a host key, the secure shell parameters must be configured.

Command:	sshd
Usage:	sshd {erase|gen} **erase** will erase the host key, if it exists. **gen** will generate a new host key and save it.
Type:	# **sshd erase** # **sshd gen**
Result:	1 st line - host key is erased 2 nd line - host key is generated
+	Must have administrative privileges to use this command.

Configuring Secure Shell Parameters

To begin the secure shell parameter configuration process, use the following procedure:

1. From a shell prompt, run the `apps` command to display a list of applications and their current enabled or disabled status.

NOTE: To secure your RCM 4/8, it is suggested that httpd and telnetd be disabled and the sshd be enabled. To leave your RCM 4/8 unsecured, it is suggested that sshd be disabled. For the time being, leave httpd and telnetd enabled so that if something goes wrong during the configuration process telnetting in provides the ability to fix it.

2. Enable the sshd applications with the following command:

```
apps set sshd enable
```

3. Type `show ssh` and the ssh parameters are displayed:

```
! SSHD Configuration
!
set ssh hostkey 1048
set ssh serverkey 768
set ssh authgrace 300
set ssh regen 3600
set ssh port 22
set ssh authmethod “Local then RADIUS”
set ssh allowroot Yes
```

Table 3: ssh Parameter Definitions

Parameter	Definition
hostkey	Number of bits that are used to generate the host key.
serverkey	Number of bits that are used to generate a server key.
authgrace	Number of seconds that a user has to successfully authenticate before being cut off.
regen	Defines the usable life span of a server key in seconds.
port	Defines the TCP port on which sshd listens for connection requests.

Table 3: ssh Parameter Definitions

Parameter	Definition
authmethod	Defines the method by which users are authenticated. Possible values are “local”, “radius”, or “both”. In the latter case, the local user file is checked first, then RADIUS.
allowroot	Either allow or disallow a user to login as root.

4. Set the parameters as desired using one or more commands of the form:

```
ssh set <paramName> <value>
```

5. Save these configuration parameters to flash so that they are used the next time the machine is booted. Saving to a TFTP host is fine as well, as long as the system is configured to fetch its configuration from that same TFTP host. To save, type:

```
save
```

or

```
save <hostname> <filename>
```

All the configuration parameters are now set.

telnet

Telnet is traditionally used to log into a remote host in order to run interactive terminal sessions. In a more general sense, telnet communicates with a host over the network using the Internet telnet protocol. This implementation also allows you to communicate with arbitrary TCP service ports, and even create a TCP connection that bypasses telnet protocol entirely by sending the data directly between your terminal and the TCP connection. This allows the ***telnet*** command to be used for purposes beyond its original intention. The following are examples of how the telnet command might be used from the IntelliServer's command line:

Command:	**telnet**
Usage:	**telnet** [<hostname | ip-address> [<port>]] [-t <termtype>] [-E] [-8] [-R[C]] [-Pn] -E no telnet escape -8 telnet binary default -R raw TCP (non-telnet) -RC raw with icanon -P[1|2|3] Return key: <default>= cr/null 1=cr, 2=cr/lf, 3=lf
Type:	# telnet 160.77.99.100 -t wy60 # telnet feather -RC # telnet jeeves 9003 -8
Result:	Line 1 - telnet to ip address 160.77.99.100 terminal type wy60 Line 2 - telnet to hostname feather with keyboard configured for canonical input Line 3 - telnet to hostname jeeves at port 9003 with Esc key disabled

Telnet Arguments and Options

Command-line arguments are used to specify which host is to be accessed and which TCP port on that host, when using telnet for non-standard connections. Command-line options are used to supply terminal names and for choosing other defaults for telnet negotiations. The following table shows the telnet command-line arguments and options.

Table 4: Telnet Command-Line Arguments and Options

Arguments…	
hostname *ip-address*	The name or IP address of the host. If a host name is used, the IntelliServer must resolve it using its host table or external name servers. When a host name or IP address is supplied, telnet immediately tries to open a connection to that host. If it fails to open a connection, the telnet command exits with an error message. (It does *not* drop into command mode, as do many implementations because this would constitute a security loophole for users configured to only telnet to certain hosts).
port	The TCP port number. The default is 23, which is the well-known port for telnet service. In some installations different services may be available on other TCP ports which telnet could access in this way.
Options…	
-t *termtype*	Allows the specification of a terminal name to the remote host. If this option is not supplied, *Remote Terminal Type* configured for your port is used. If *that* is blank, then the *Local Terminal Type* is sent. The host sets up the user's TERM environment variable based on the terminal type the IntelliServer sends. This enables screen-based applications to run properly. In order to be useful, the terminal name specified must correspond to an appropriate entry in your host's */etc/termcap* file or *terminfo* database.
-E	The telnet *escape* character is used for switching telnet into command mode. By default, this character is *ctrl-***]**. When the **-E** option is used, the regular escape key is treated like an ordinary character and the telnet *escape* function is performed by the terminal's **Break** key instead.
-8	This disables the telnet *escape* key in the same way as the **-E** argument but also negotiates with the host to switch to *telnet binary* mode. If the **-8** option is used, the **-E** option is superfluous.

Table 4: Telnet Command-Line Arguments and Options (Continued)

-R **-RC**	These are usually used only when a TCP port (other than 23) has been specified and disables telnet protocol entirely. Raw data from the keyboard is sent to the TCP connection and raw data from the connection is sent to the terminal. When the **-RC** option is used, the keyboard is configured for canonical input; otherwise, the keyboard input is raw. Since *telnet* protocol is not used, there is no telnet escape key and no command mode. The **break** key from the terminal terminates the session. These options were intended to enable the IntelliServer to access simple custom applications you might write.	
-P1 -P2 -P3	These options are used to change the way *Carriage Return* (*hex 13)* characters are handled when you are not operating in telnet binary mode. This affects only characters that are *received* by the serial port (normally this would be keyboard data). This is provided for compatibility with hosts with archaic telnetd implementations.	
	default	According to *telnet* standard - *Carriage Returns* are padded with nulls (0).
	-P1	No null padding after *Carriage Returns.*
	-P2	*Newlines (0x10)* added after *Carriage Returns.*
	-P3	*Carriage Returns* replaced by *Newlines.*

Using Telnet Connections

When a telnet connection is open, data from the keyboard is sent to the remote host and data from the remote host is sent to your terminal. When a TCP port number that is not specified, connect to the default telnet service port, 23. On the other end of your connection there will be something there to log into the host system and offer the appropriate services based on your login.

On UNIX hosts, this service is usually performed by a process called *telnetd* which creates a '"pseudo-tty" device on the host system and then sends data through this device to reach the application.

tip

The ***tip*** command allows a user to connect to a serial port.

Command:	tip
Usage:	+tip <port number>
Type:	# **tip 1**
Result:	A tip connection is established with port 1.

tty

The ***tty*** command shows the user what port is on.

Command:	tty
Usage:	# tty
Type:	# tty
Result:	tty 24 session 0

version

The ***version*** command displays the release and version numbers of the IntelliServer software that is currently running. It is important that you know the release number if you are needing help from Computone's technical support department.

Command:	version
Usage:	# version
Type:	# version
Displayed:	Computone RCM 4/8 Intelliserver Release 1.4.005 Version 20001027

whodo

The ***whodo*** command lists the commands that are running on all active serial ports, as well as administrative sessions created when you telnet into the IntelliServer. These sessions are designated by the port names *pts0* and *pts1*, while the serial ports are designated by their port numbers. Active PPP, SLIP, and CSLIP connections are shown, as well as active connections to *Reverse-TCP* and *Printer* ports.

Command:	**whodo**
Usage:	whodo whodo all
Type:	# whodo # whodo all
Result:	Line 1 - whodo pt-ses day time owner command 0 0 007 00:20 root shell pts00 0 007 00:38 root whodo Line 2 - whodo all pt-ses day time owner command 0 0 007 00:21 root shell 1 0 000 19:01 root init awaiting DCD 2 0 000 19:01 root init awaiting DCD 3 0 000 19:01 root init awaiting DCD 4 0 000 19:01 root init awaiting DCD 5 0 000 19:01 root init awaiting DCD 6 0 000 19:01 root init awaiting DCD 7 0 000 19:01 root init awaiting DCD 8 0 000 19:01 root init awaiting DCD 9 0 000 19:01 root init awaiting DCD 10 0 000 19:01 root init awaiting DCD 11 0 000 19:01 root init awaiting DCD 12 0 000 19:01 root init awaiting DCD 13 0 000 19:01 root init awaiting DCD 14 0 000 19:01 root init awaiting DCD 15 0 000 19:01 root init awaiting DCD pts00 0 000 00:00 root whodo

The ***whodo all*** command lists everything ***whodo*** does, but also gives the status of any login ports which are enabled but not yet active (i.e., those waiting for carrier or for a response to login). These commands are useful for providing an overview of what is happening on each port of the IntelliServer. For example, you might use this command to see if there is anything important going on before shutting down the IntelliServer.

CHAPTER 7

Remote Control Management

Remote control management is a software package based on a client/server technology. The software package is designed to provide the monitoring of all devices attached to the IntelliServer's ports. The user has the capability to monitor IntelliServer's ports from a single remote location via the internet. It is designed to be used with any operating system and comes with a Java environment provided if necessary. The IntelliServer must be configured before using this application.

The following table lists the management features available.

Table 1 Management Features

Features	Description
Login Dialog Box	Allows access to the management program.
Main Panel Window	Allows access to the various options available in the program.
Monitoring Widow	Allows the monitoring of the selected location IntelliServer.
Manual Connection Dialog Box	Allows a connection to a known host IP, port and server.
Configure Dialog Box	Allows the user to send commands to the connect port.
Administration Window	Allows the user to add, delete, edit, save to a file, and change a password.

Login Dialog Box

Selecting the ***Remote Control Management*** program initiates the program and the ***Login*** window appears.

Figure 1 Login Window

Enter the ***Login Name*** and ***Password*** and click ***OK.*** The Main Panel window appears:

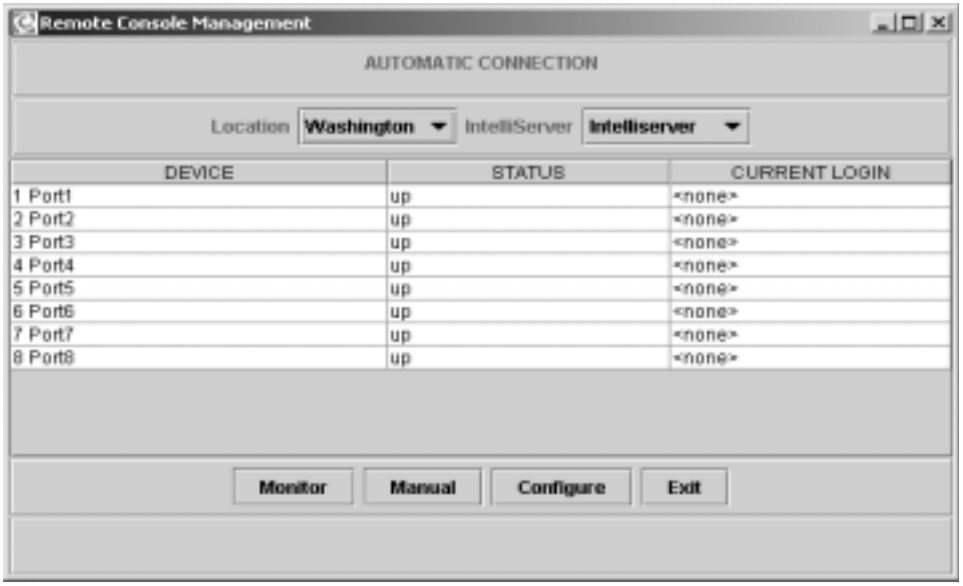


Figure 2 Main Panel Window

Main Panel

This screen allows the automatic connection to the IntelliServers configured at the selected location. The options available are:

- ***Location***- Select the desired geographic location of the IntelliSever.
- ***IntelliServer***- Select the desired InteeliServer.
- ***Monitor***- Selects the Monitoring screen.
- ***Manual***- Selects the Manual Connection dialog box.
- ***Configure***- Selects the Administration window.
- ***Exit***- Exits current screen.

Monitor

Selecting ***Monitor*** button on the main panel window displays the following screen:

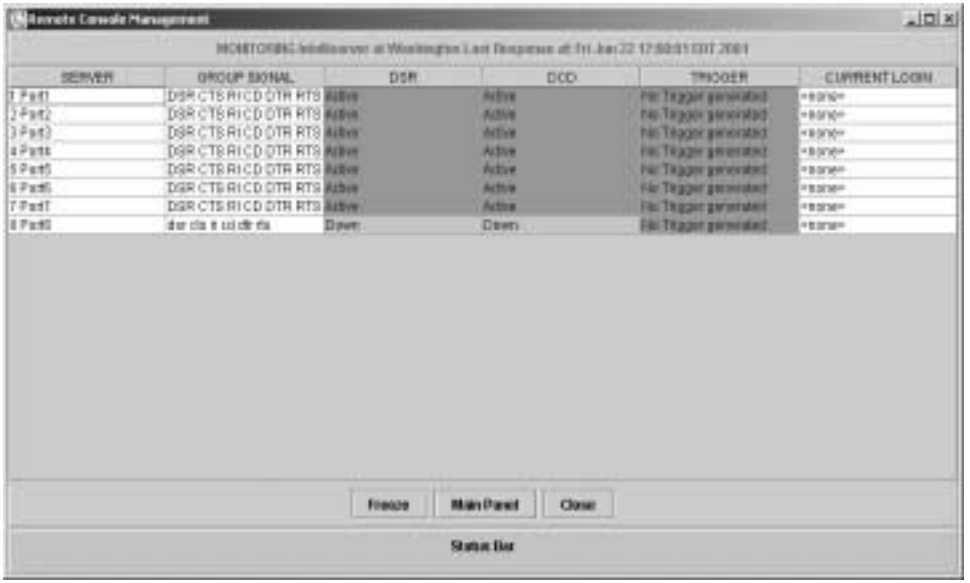


Figure 3 Monitoring Screen

This screen allows the user to monitor the port status of the selected IntelliServers configured.

NOTE: There is a slight time delay before the data is displayed. This delay is due to the connection time and the reception of return data.

The options available are:

- ***Freeze*** -This freezes the current displayed screen, but the monitoring continues and another screen will appear in the background.
- ***Main Panel*** - Returns to the Main Panel display.
- ***Close*** - Closes the program.
- ***Status Bar*** -The Status Bar displays information such as waiting for data, and error messages.

Manual

Selecting ***Manual*** button on the ***Main Panel*** window displays:

Figure 4 Manual Connection Window

Typing in the *Host IP*, *Port No.* and *Server*, a direct connection can be made to the host.

Click ***Connect*** and the ***Terminal Emulator*** window appears.

Click ***Main Panel*** and the ***Main Panel*** window appears.

Terminal Emulator

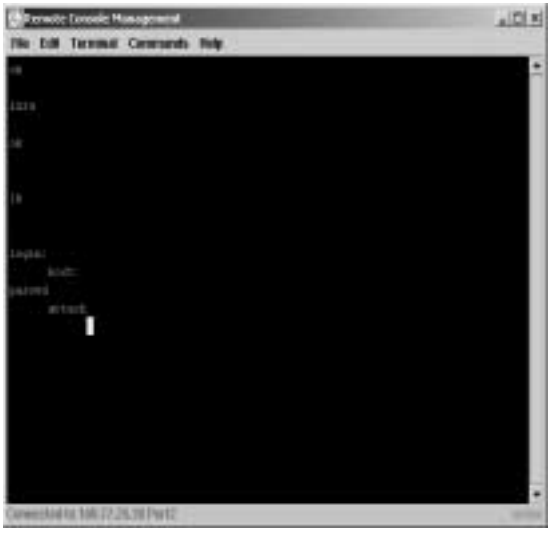


Figure 5 Terminal Emulator Window

Typed commands can be made to the attached host.

Configure

Selecting the ***Configure*** button on the ***Main Panel*** displays the following window:

Figure 6 Administration Window

Access to the ***Administration*** window options requires a password. A dialog box appears for entering a password.

The options available are:

- ***Add Locations*** - Allows a new location to be added.
- ***Delete Location*** - Deletes a current location.
- ***Edit entry*** - Adds a new row to the window to allow the addition to the existing location.
- ***Save to File*** - Save the data to a file.
- ***Change Password*** - Allows the current password to be changed.
- ***Monitor List*** - Provides a list of all servers on the system from each location which require monitoring.
- ***Main Panel*** - Returns to the ***Main Panel*** window.

- ***Double Click*** - Double click an IntelliServer to go to the ***Terminal Emulator*** window.

INDEX