TOTAL CONTROL™

NETServer 3.1

Primary Rate ISDN NAC

RELEASE NOTES

8100 North McCormick Blvd.

Skokie, IL 60076-2999

Table of Contents

What's New in Release 3.1

Release 3.1 of the NETServer firmware is divided into two variants: a standard release and an enhanced release (NETServer PRI). NETServer PRI supports all features of the standard release, plus some additional functions. Although this document is primarily concerned with these additional functions, the new features in the standard release will be mentioned first. Coverage of the additional features in NETServer PRI begins on page 8.

NETServer 3.1 standard contains the following enhancements to the 3.0 code:

- Classless InterDomain Routing and Host-based routing via the Netmask Table.
- IP address spoofing.
- UNIX, Total Control and / or RADIUS accounting servers, ANI / DNIS, and ICMP error message logging.
- SLIP / PPP processing by modems, modem hot swap support, and modem initialization strings.
- Multiple Gateway Cards (other NETServers, X.25 PAD, API Card) in a single chassis.
- Multiple name server support.
- Randomized use of Default / Alternate Hosts for load balancing.
- Modem port enhancements.

Additional Software Enhancements

- 60 port support. The NETServer software no longer limits you to a 24 or 48 port license. Any NETServer may now support up to 60 ports.
- NetBIOS over IPX support
- PAP enable / disable
- Netdata supports both TCP / IP and SPX / IPX
- Pre-allocated system netbufs increased from 1000 to 1400
- Rezero network statistics and session statistics saved until next call

- Unidirectional Van Jacobson compression
- Users set to Prompt may specify a TCP port with the host name or IP address when using Telnet

Netmask Table

CIDR (Classless Interdomain Routing) or host-based routing requires special netmasks. Special netmasks may also be useful for debugging.

The Netmask Table allows you to configure netmasks for CIDR or host-based routing as needed. RIP messaging / dynamic route information must be active for host-based routing.

IP Address Spoofing

The NETServer may now be configured to spoof a single IP address. When the NETServer identifies itself to remote routers or other remote devices, it uses this IP address rather than the IP address of its LAN interface.

IP address spoofing is useful when more than one NETServer must appear to be a single router or other device to remote networks and other routers.

New Modem Features

The NETServer now supports the following features:

- modem hot swap
- modem init string generation
- modem processing of SLIP or PPP frames

Modem Hot Swap

Previously, if a Quad Modem Card was removed from a slot with an active packet bus connection to one of its modems and then reinserted, the modem had to be manually reinitialized. The packet bus connection also had to be manually restored from either the Command Line or the Windows Management software.

With Release 3.1 of the NETServer, Quad Modem Cards that are removed from a slot with an active packet bus connection and then reinserted are automatically reinitialized and the packet bus connection restored.

Modem Init Strings

A new software table allows you to create customized modem initialization strings. You may then assign the modem init strings to one or more modem ports.

When a modem port is reset, the init string is sent to the Quad Modem Card modem corresponding to that modem port. If the modem itself resets after a disconnect, the init string is also sent.

SLIP/PPP Processing in Modem

Release 3.1 of the NETServer supports Quad Modem Cards with SLIP and/or PPP packet processing capability. Note that this applies to Quad Modem Card code releases 2.0.4 and later *only*.

You can now configure the NETServer to permit Quad Modems to perform the initial PPP or SLIP packet processing. This increases performance by freeing up processing resources in the NETServer for other tasks.

Accounting Server

The NETServer supports the following new features:

- Log events to a UNIX or Total Control accounting server
- Log accounting information to a RADIUS accounting server
- ANI and DNIS call information
- Log ICMP error messages to a syslog server

Accounting Server Support

The NETServer now supports event logging. You can configure the NETServer to send event information to a Total Control Accounting Server or a UNIX accounting server. You can also configure the NETServer to send the event information to an alternate accounting server if the primary server is unavailable.

Event logging is performed by transmitting a record containing event information from the NETServer client to an accounting server. TCM uses the RADIUS client/server model for this feature.

RADIUS Accounting and ANI/DNIS

Release 3.1 of the NETServer supports the current RADIUS Accounting Internet Draft. The NETServer can generate appropriate Code 4 Accounting-Request and Code 5 Accounting-Response messages for properly configured RADIUS servers.

The NETServer's RADIUS implementation also supports ANI and DNIS services.

ICMP Message Logging

If your system uses syslog network accounting, you can configure the NETServer to send ICMP error messages to the syslog server.

Multiple Gateway Cards

With Release 3.1 of the NETServer, users can install multiple Gateway Cards in a single chassis. This new feature allows the system administrator to configure different applications for modems in the chassis.

For instance, a user can configure a NETServer card to address modems 1-24, an API card to address modems 25-28, and a X.25 PAD to address modems 29-48 within a single chassis. You can also install two NETServers in a chassis (NETServer 1 addressing modems 1-24 and NETServer 2 addressing modems 25-48). This load sharing improves throughput on a per-port basis, and provides redundancy.

Note: It is important that you do not configure two Gateway Cards to use the same modem, or call routing may be haphazard. Future management releases will provide a modem configuration check, but it is currently up to the user to avoid redundant configuration.

Multiple Name Servers

Release 3.1 of the NETServer supports up to two name servers. The first is a primary name server, and the second is a backup server that is used when the primary name server is unavailable.

Note: The NETServer does not support more than one name service at a time (DNS and NIS cannot both be running).

Randomized Hosts

You can now relieve the burden on frequently-used global default, port default and RADIUS user table hosts, by randomizing the selection of the host chosen for user sessions. When this feature is enabled, a preferred host will be randomly chosen from among the default and alternate hosts defined rather than always preferring the default host.

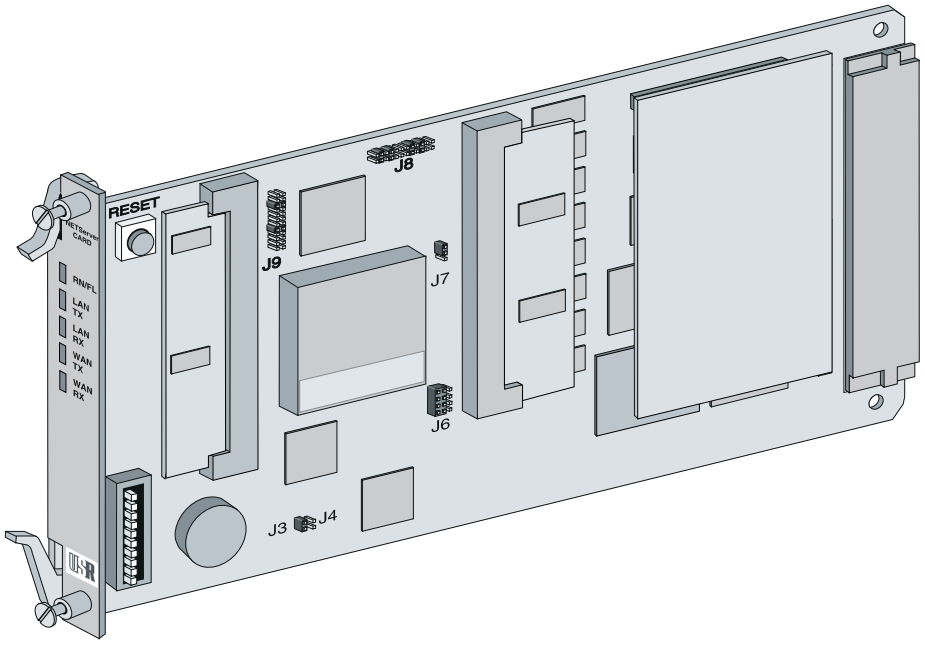
New Modem Port Features

Release 3.1 of the NETServer Command Line and NETServer Manager software now support the following modem port features:

- Detect and flush of stopped ports
- Dialback delay
- Port status display shows current and configured status
- Ports reset if Carrier Detect is lost before a user connects to a host
- Support for up to eight Alternate Hosts

NETServer PRI

Affectionately known as the “kitchen sink” NETServer, the NETServer PRI card can do everything the standard NETServer platform can do. Additionally, it supports primary rate ISDN dial in, two frame relay interfaces, miscellaneous enhancements to global configuration, and the ability to negotiate the local IP address of a dial in user. NETServer PRI also contains additional SNMP MIB support, allowing basic monitoring and debugging of the NETServer to be performed in Total Control Manager.



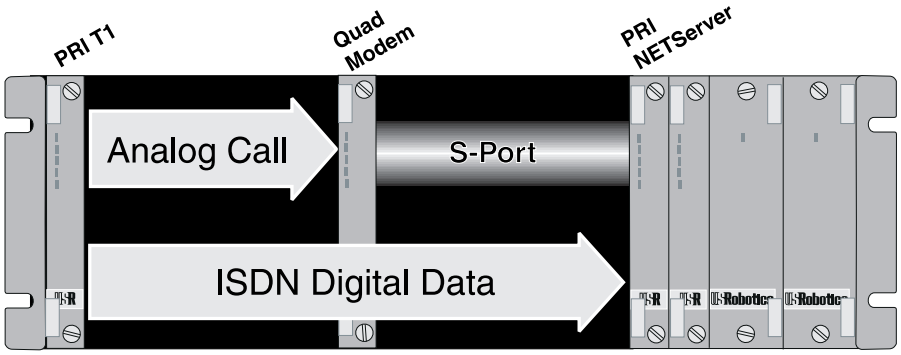
Additional Hardware Requirements

The NETServer PRI firmware requires three hardware additions to operate correctly:

- a high speed Ethernet/frame relay NIC
- 4 megabytes of extra RAM (SIMM)
- a special ISDN daughter card

Primary Rate ISDN

When used in conjunction with a primary rate ISDN T1/E1 card, NETServer PRI can respond to both voice grade (modem) and ISDN digital calls.



PRI NETServer and T1/E1 Card in Action

Calls signaled as voice grade audio are forwarded to a digital quad modem the same as they are with a non-ISDN NETServer and T1 /E1.

However, when a call is signaled as ISDN data, the data is forwarded directly to the NETServer (over the TDM bus). Because there is no modem involved in this type of call, there is no S-port that can be configured beforehand. The NETServer dynamically allocates bandwidth for each ISDN call and frees those resources when the call is torn down.

From a configuration standpoint, this means that any quad modems used for a given application will still require S-port setup. However, the temporary ports created for ISDN service are essentially configuration free. To configure ISDN applications, you need only create user table entries for users and networks that will be dialing in over ISDN.

Note: The initial release of NETServer PRI supports only dial in on ISDN lines. It does not support ISDN dial out .

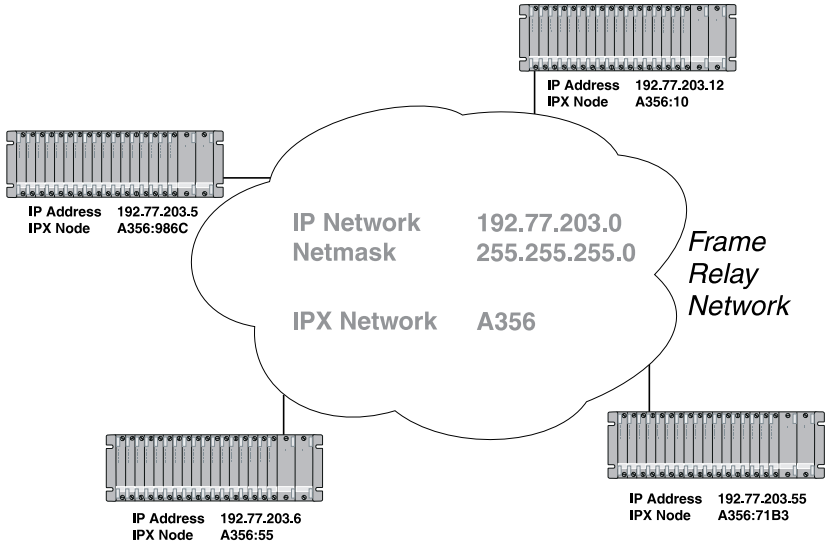
Although ISDN service should not require any additional configuration, there are two additional tables that can be used to fine tune ISDN service: the service profile table and the call mapping table. These tables are used to override faulty D-channel signaling and therefore are not needed unless you are receiving improperly signaled calls. Since this is an

extremely rare circumstance, you will probably never have to use these tables. See Appendix F in the NETServer 3.1 Command Reference for more information.

Frame Relay Configuration

Frame Relay support is no longer a special release of the NETServer firmware. It is now a standard component of NETServer PRI. NETServer PRI ships with an Ethernet / Frame Relay combination NIC rather than the Frame Relay only NIC that came with the 3.0 Frame Relay NETServer. This arrangement allows the frame relay NETServer to be managed over a local Ethernet network.

As with version 3.0, configuration of each frame relay port is a matter of defining a subnet comprised of all nodes connected to that port via frame relay virtual circuits. This makes the frame relay network transparent to the TCP / IP and IPX protocols.



For example, to configure a NETServer (in this case, the one in the upper left corner) whose WAN0 port is connected to the subnet pictured above, you would use the following commands:

```
set wan0 address 192.77.203.5
set wan0 netmask 255.255.255.0
set wan0 IPX network A356
```

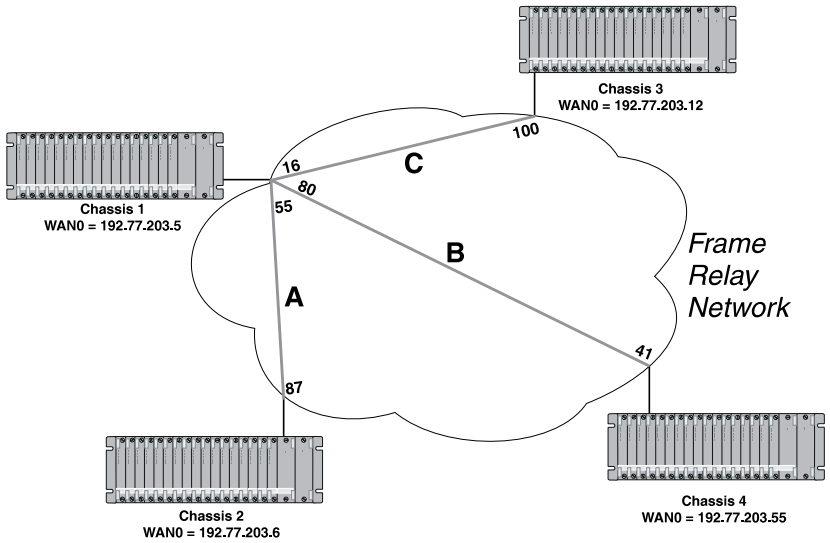
Once you create this subnet, you will have make sure that all DLCI numbers are mapped to the appropriate IP and IPX node addresses on the other end of the corresponding virtual circuits. Most IP networks will be able to inverse ARP for this information. For IPX networks and IP nodes which do not support inverse ARP, however, you will have to create static mappings.

In version 3.0, you had to map all DLCIs with a single *set dlci...* command. Version 3.1 makes the process a little bit easier by changing this command to *add dlci...* which allows you to define DLCIs as they are added to your network rather than all at once. 3.1 also features a *delete dlci...* command. The actual syntax for the add command is shown below:

add dlci <*wan0* | *wan1*> <DLCI #>**:**<IP addr> <DLCI#>**:**<hex IPX node (MAC) addr>

For example, if you wished to configure the NETServer in the upper-left corner of the drawing on the previous page, it would probably be connected to each of the other three NETServers by a different permanent virtual circuit. To configure virtual circuit "C", you could use the following command:

add dlci wan0 16:192.77.203.12



If you go back to the previous illustration, you can see that these NETServers are also routing IPX. So, you would also include the IPX node address.

add dlci wan0 16:192.77.203.12 16:10

Another difference in 3.1 PRI's handling of frame relay is that the syntax of the *show* command has been changed to more accurately reflect what the commands actually do. A summary of the changes is in the table below:

3.0 frame relay command	*3.1 PRI command*
show <*wan0* | *wan1*> **interface**	**show** <*wan0* | *wan1*> **mib interface**
show <*wan0* | *wan1*> **dlci**	**show** <*wan0* | *wan1*> **mib pvc**
show <*wan0* | *wan1*> **errors**	**show** <*wan0* | *wan1*> **mib errors**
show <*wan0* | *wan1*> **lmi**	**show** <*wan0* | *wan1*> **dlci**

For more information about frame relay configuration for NETServer PRI, see Appendix G of the NETServer 3.1 Command Reference.

Changes to Global Configuration

Although most of NETServer PRI's informational displays are identical to those of the standard Ethernet or Token Ring NETServer, global configuration changes significantly. When you type *show global* on a NETServer PRI, you will see a screen similar to the one below.

Compare this to the *show global* display at the beginning of Chapter 10 in the NETServer 3.1 Command Reference, which shows the information displayed by a standard NETServer.

```
        System Name:  Cincinnati

      Default Hosts:  192.77.203.54        192.77.203.55        192.77.203.64
                      192.77.203.65

         IP Gateway:  192.77.203.12        Gateway Metric:  1
        IPX Gateway:  00000000:000000000000  Gateway Metric:  1
      Default Route:  Quiet (Off)

             Domain:  My_Domain             Name Service:  DNS
       Name Servers:  192.77.203.13         0.0.0.0        0.0.0.0

       Sys Loghosts:  0.0.0.0               0.0.0.0
       RADIUS Server: 192.77.203.100        Alternate Server:  0.0.0.0
    Accounti ng Server: 192.77.203.100      Alt. Acct. Server: 0.0.0.0

    Telnet Access Port:  23
      Assigned Address:  192.77.203.101     Reported Address:  0.0.0.0

        PPP in modem:  OFF   SLIP in modem:  OFF   Packet Bus Clock:  Master
    ICMP error Logging:  OFF   Connect Message:  ON   Dial !root Access:  ON
      Random hosts list:  OFF   SNMP:  OFF   Proxy Arp:  OFF

  DNS cache reset timeout:  0 days  0 hours  30 minutes  (30 min)
  Configured Ethernet media:  None
  Currently Active B-channels:  12   Maximum Active B-channels:  60
```

NETServer PRI "show global"

For clarity's sake, the above illustration highlights the items which differ from the standard NETServer display. The differences are documented (in alphabetical order) below. See the NETServer 3.1 Command Reference for information on other items.

Configured Ethernet Media

Previous versions of the NETServer firmware automatically detected which type of Ethernet cable was connected to the NIC. Although convenient, auto-detection has two disadvantages:

- There is a slight delay at boot time (while auto-detection takes place).
- If you don't attach an Ethernet cable to any of the interfaces, lights flash at you and you get a lot of annoying messages in debug mode.

NETServer PRI allows you to specify what type of cable is being used.

set media <*none* | *autodetect* | *10baset*>

none	Assume that no cable is attached. Suppress LED flashing and related error messages.
autodetect	(default) Autodetect Ethernet cable type.
10baset	Assume 10baset connection.

Note: The firmware also supports the keywords *10base5* and *10base2*. However, the NIC that NETServer PRI is currently used with does not have such interfaces.

Currently Active B-channels

This displays the number of ISDN B-channels which are currently being used for active sessions.

DNS Cache Reset Time-out

Once the NETServer has obtained a name resolution response from a DNS server, it caches the results so that the name resolution information can be reused without further DNS requests. The following command configures the length of time NETServer caches individual DNS responses before refreshing them (by issuing another DNS request):

set dnscache <DD:HH:MM>

You must specify days, hours and minutes as three two-digit pairs separated by colons. For example,

set dnscache 00:00:30

The default is 30 minutes. Earlier versions of the NETServer firmware also used 30 minutes for this function, but did not allow you to change it.

IPX Gateway

NETServer PRI displays the IPX default gateway and metric. Previously, only the IP default gateway had been displayed.

Maximum Active B-channels

This value displays the maximum number of B-channels that the NETServer may use at any one time. As documented on page 2-15 of the (NETServer 3.1 Command Reference, the value can be configured with the following command:

set maxb <0 .. 60>

Name Server 3

NETServer PRI allows you to specify a third name server using the following command:

set nameserver 3 <IP address>

NETServer PRI checks with each name server in turn. If no valid response is received in the amount of time given below, NETServer proceeds to the next name server.

Nameserver 1	5 seconds
Nameserver 2	5 seconds
Nameserver 3	15 seconds

If no valid response is received from Nameserver 3, name resolution fails.

Note: On all previous versions of the NETServer firmware, the time-out for Nameserver 2 was 15 seconds.

Proxy Arp

Proxy ARP can be useful when an ARP request is received on one interface, but the NETServer knows that the requested address is actually reached through another interface. Since the requested address is not actually on the subnet from which the ARP originated, there is no way that the desired machine can respond.

When proxy ARP is enabled, the NETServer will reply to such a request with its own MAC address. Packets headed for the requested system will then be sent to the NETServer, which in turn forwards them to the appropriate address.

set proxy <*on* | *off*>

Random hosts list

This displays the status of the *Randomize Hosts* setting as described on page 10-4 of the NETServer 3.1 Command Reference.

SNMP

This field indicates whether SNMP operation has been enabled or disabled. See *SNMP configuration* in Chapter 10 of the Command Reference for more information.

Syslog Host 2

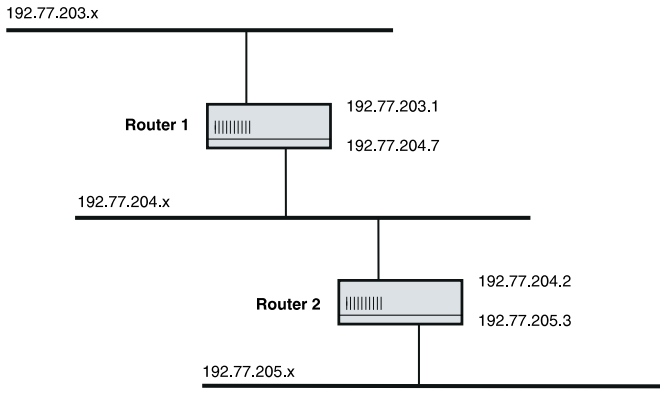
NETServer PRI allows you to send Syslog accounting messages to two different Syslog hosts simultaneously. Type the following command to add the second host:

set 2ndlog <IP address>

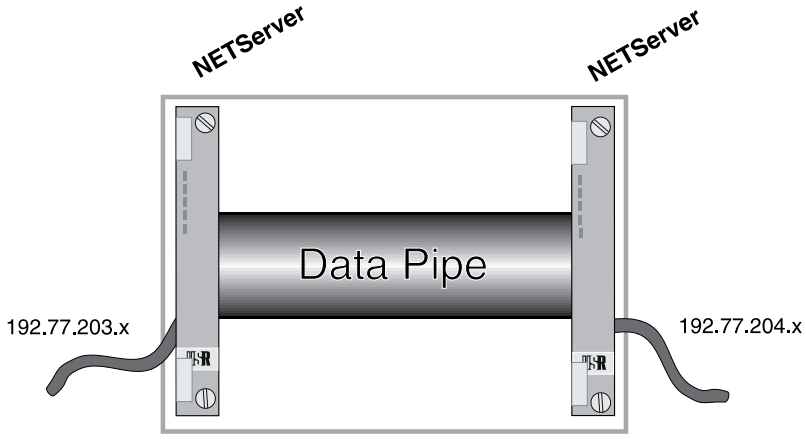
Note: The two loghosts are independent of one another. NETServer can send accounting messages to the secondary loghost even if the primary loghost is not defined.

IP Addressing for S-Ports

Traditionally, each interface of a TCP/IP routing device is assigned a different address. This allows the device to operate as a node of each subnet it is connected to.



Like many dial up routers, both the standard NETServer and NETServer PRI simplify this structure for the sake of efficiency. Individual IP addresses are not required for its modems. Instead, the NETServer prefers to define a dial up connection as a data pipe between two routing devices. Essentially, the NETServer and the remote routing device act like two halves of a single router.



Two routers acting as a single device

This data pipe approach to dial up connections has several advantages:

- Since IP addresses are not actually used for the modem ports, address space can be conserved for other uses.
- It avoids the problem of defining an entire subnet for each point-to-point link. After all, such links contain only two nodes. The next hop is guaranteed to be the device on the other end of the link.
- When talking to other NETServers, several modems can be used as part of the same data pipe and yet the NETServer will see only a single IP address on the other end of the connection. This is much simpler than trying to divide packets between multiple parallel IP subnets.

Negotiating an S-port address

Although the approach described above works well with other NETServers and some third party routers. Other routers may insist on seeing addresses assigned to the NETServer's modems. For this reason, NETServer PRI can be configured to negotiate the local address of a dial in connection. The command to do this is

set user <user name> **address negotiate-both**

- or -

set user <user name> **address negb**

When such a user dials into the NETServer, the NETServer will use IPCP negotiation to obtain the addresses on *both* ends of the connection.

NETServer PRI MIBs

NETServer PRI contains MIB support not present in earlier versions of the NETServer firmware. These MIBs may be viewed by clicking on the appropriate NETServer card in Total Control Manager and then selecting Programmed Settings (from the Configure menu) or Trap Settings (from the Faults menu).

Programmed Settings

NETServer Identification

Operational Status *(RO)*

MIB Object: uchasEntityOperStatus

Description: Displays the current operational status.

Settings:
Other
Out of Service
Testing
Operational
Failed
Loading

Serial Number *(RO)*

MIB Object: uchasSlotModuleSerialNumber

Description: The Card's hardware serial number, as stored in EEPROM.

Settings: Display String

Hardware Revision *(RO)*

MIB Object: uchasSlotModuleVersion

Description: The Card's hardware revision level, as stored in EEPROM.

Settings: Display String

Software Version *(RO)*

MIB Object: uchasEntityVersion

Description: The revision level of the software being executed by the Card's processor.

Settings: Display String

***DIP Switch Settings on the NAC** (RO)*

MIB Object: uchasSlotSwitchSettings

Description: Displays the current DIP switch settings on the selected NAC.

Settings: Display String

***DRAM Installed on the NAC** (RO)*

MIB Object: uchasSlotRamInstalled

Description: The number of Kbytes of DRAM on the Card.

Settings: 0..4294967295

***ROM Installed on the NAC** (RO)*

MIB Object: uchasSlotFlashInstalled

Description: The number of Kbytes of FLASH ROM on the Card.

Settings: 0..4294967295

Packet Bus Sessions

The Packet Bus Sessions configuration group contains parameters that assist the network administrator in configuring dynamic packet bus sessions.

Slot Session Assignment

MIB Object: pbSessionDestSlot

Description: Identifies the slot to which a packet bus session has been assigned.

Settings: 1..17

Channel Session Assignment

MIB Object: pbSessionDestChan

Description: Identifies a particular entity to which a packet bus session has been assigned.

Settings: 1..48

Availability of Row for Packet Bus Session

MIB Object: pbSessionRowState

Description: Indicates if a packet bus row is available for assignment to a packet bus session.

Settings: free
used

Default: free

Session Assignment between Entities

MIB Object: pbSessionReqStatus

Description: Used to assign or delete a session between entities specified by the table indices. Connected means a packet bus session link start has been requested, and disconnected means a packet bus session link termination has been requested.

Settings: disconnected
connected

Default: disconnected

Session Status

MIB Object: pbSessionStatus

Description: Displays the current status of the packet bus session. A status of Unassigned coupled with a state of used indicates that a packet bus failure has occurred.

Settings:
unassigned
assigned
connected

Last Packet Communication Type

MIB Object: pbSessionLastRequest

Description: This parameter indicates the packet type last sent over the packet bus.

Settings:
unknown
open
close
listen
dial
disconnect
transmit
receive
setMode
query
flush
kill
reserve
answer
attach
disconnected
connected

Session Tx Packet Count

MIB Object: pbSessionPktSent

Description: A counter that indicates number of packets that have been sent.

*Setting:*Counter

Session Rx Packet Count

MIB Object: pbSessionPktRcvd

Description: A counter that indicates number of packets that have been received.

*Setting:*Counter

Session Packet Size

MIB Object: pbSessionPktSize

Description: A number that indicates the packet size of the current or last established session.

Settings: 0..4294967295

Default: 0

Session Packet Timeout Count

MIB Object: pbSessionBusTimeOut

Description: A counter that indicates number of packet bus timeout that have occurred.

*Setting:*Counter

Session Error Status

MIB Object: pbSessionErrorStatus

Description: This parameter indicates the error status of the packet bus session.

Settings:
noError
invalidParm
socketNotOpened
noMoreSocket
connectionExist
connectionfailed
noMoreConnObj
noActiveConn
ackWaitTimeout
hwNakRcvd
otherBusError
linkStartRcvd
outOfSeqFrame
noMemory
nullPointer
invalidBlock
notInitialized
failedToRecv
invalidMsgType
exceedMaxSends
connectionReset
socketClosed
uiReqPending
heartbeatTimeout
remoteBusy

localBusy
noResponse
linkdownNoTx
nodataToTx
txPreAck
txTardyAck
txBusTimeout
rxBusTimeout
txTAL
rxTAL
txMasterTimeout
clkVanished
clkReturned
shutdown
frameerror
xIDTimeout
recvLSinInfoTransferState
recvIFrameWithWrongSeq
rxMsgBufferOverflow
linkDown
listeFailed
lisenRcvFailed
dtrDrop
answerFailed
openFailed
closeFailed
readFailed
writeFailed
autoParityFailed
setmodeFailed
badDataBase
padStreamsError
padError

NETServer Configuration

When you select this parameter group, the *NETServer Manager* configuration program is launched.

Faults

Packet Bus Traps

Packet Bus Active Session Trap

Description: Enables a trap when a packet bus session becomes active.

Settings: enable
disable

Default: disable

Packet Bus Congestion Trap

Description: Enables a trap when a valid packet bus session is in progress but either the Modem or the Gateway packet bus driver is experiencing a congestion problem.

Settings: enable
disable

Default: disable

Packet Bus Session Lost Trap

Description: Enables a trap when a valid Packet Bus session has been lost.

Settings: enable
disable

Default: disable

Packet Bus Session Inactive Trap

Description: Enables a trap when a request is made to change a packet bus session from the active to inactive state.

Settings: enable
disable

Default: disable

Packet Bus Session Error Trap

Description: Enables a trap when an error occurs during a packet bus session.

Settings: enable
disable

Default: disable

Chassis Awareness

To provide the functionality required in supporting ISDN, NETServer PRI must be aware of the current chassis NAC configuration. Toward this end, the NMC has expanded the device discovery process to detect NACs that require chassis awarenesss, to provide them with current chassis NAC configuration information to them when appropriate, and to query them for their current chassis NAC configuration when needed.

NETServer PRI is capable of storing the chassis NAC configuration, saving it to NVRAM, and restoring it from NVRAM when commanded to do so by the NMC.

Known Bugs

"Limited Expansion Room - Packet Lost"

This message will occasionally be seen when you manually dial a remote location with the *dial <location> -x* command. The message is entirely erroneous. The connection proceeds normally.

Unexpected Reset

When you issue the command below, NETServer PRI will reboot.

set all login network <*dialin* | *dialout* | *twoway*>

It is therefore necessary to configure ports that support both user login and network services individually rather than using the *set all...* command.

!rootaccess and RADIUS

As documented in Chapter 9 of the NETServer 3.1 Command Reference, the *!rootaccess* command is supposed to prevent access to the supervisor account (!root) through modem ports.

However, when RADIUS security is used, the NETServer will allow !root login through modem ports even if *!rootaccess* is off.

Other Notes for 3.1

NETServer Manager

The information in some NETServer Manager windows, notably the hosts table and the user table is obtained from the NETServer's flash memory rather than the RAM work area that you use when you configure the box. If you exit from one of these windows and then return to it, you will only see configuration items that were configured prior to your last save to flash. This does not mean that unsaved items are lost, just that you won't see them until they have been saved to flash.