MAX 200Plus 6.0.0 Addendum

Ascend Communications, Inc.
Part Number: 7820-0288-005
For Software Version 6.0.0

March 11, 1998

Ascend Customer Service

You can request assistance or additional information by telephone, email, fax, or modem, or over the Internet.

Obtaining Technical Assistance

If you need technical assistance, first gather the information that Ascend Customer Service will need for diagnosing your problem. Then select the most convenient method of contacting Ascend Customer Service.

Information you will need

Before contacting Ascend Customer Service, gather the following information:

- Product name and model.
- Software and hardware options.
- Software version.
- Service Profile Identifiers (SPIDs) associated with your product.
- Your local telephone company's switch type and operating mode, such as AT&T 5ESS Custom or Northern Telecom National ISDN-1.
- Whether you are routing or bridging with your Ascend product.
- Type of computer you are using.
- Description of the problem.
- How to contact Ascend Customer Service

After you gather the necessary information, contact Ascend in one of the following ways:

Telephone in the United States	800-ASCEND-4 (800-272-3634)
Telephone outside the United States	510-769-8027 (800-697-4772)
Austria/Germany/Switzerland	(+33) 492 96 5672
Benelux	(+33) 492 96 5674
France	(+33) 492 96 5673
Italy	(+33) 492 96 5676
Japan	(+81) 3 5325 7397
Middle East/Africa	(+33) 492 96 5679
Scandinavia	(+33) 492 96 5677
Spain/Portugal	(+33) 492 96 5675
UK	(+33) 492 96 5671
Email	support@ascend.com
Email (outside US)	EMEAsupport@ascend.com
Facsimile (fax)	510-814-2312
Customer Support BBS by modem	510-814-2302

You can also contact the Ascend main office by dialing 510-769-6001, or you can write to Ascend at the following address:

Ascend Communications
1701 Harbor Bay Parkway
Alameda, CA 94502

Need information about new features and products?

Ascend is committed to constant product improvement. You can find out about new features and other improvements as follows:

- For the latest information about the Ascend product line, visit our site on the World Wide Web:

  `http://www.ascend.com`

- For software upgrades, release notes, and addenda to this manual, visit our FTP site:

  `ftp.ascend.com`

Contents

Ascend Customer Service iii

Introduction 1

What is in this addendum 1
Related publications 1
A few words about this release 2
Upgrading system software 3
Known issues 17

WAN access features 18

MAX CH CNT upper limit specific to platform 18
B-Channel preference when MP+ or BACP adds bandwidth 18
BACP/BAP PPP protocol IDs match IETF 19
TACACS+ server retry attempts 19
Set Cause Code for ISDN DISCONNECT 19
Raw TCP connection enabled 21
User-definable TCP connection retry timeout 23
MAX supports new CSLIP Auto Detect parameter 25
More information provided for SLIP connections 25
Multilink or MP+ call now can span multiple MAX units 27
Updated Microsoft Callback Control Protocol support 34
MAXDial/IP support for immediate modem call restriction 37
Support added for multiple host selection 38
Limiting terminal server access per user 41
User-configurable call blocking after failed connection attempt 43
Specifying Shared Profiles per Connection Profile 44
Terminal server users can be forced to use unique profiles 45
CR-LF characters act as a single CR in terminal server 46

IP routing features 46

Changes to the IP router 46
Alphanumeric IP pool names 49
Local DNS host address table option added 50
UDP Queue Control 55
Specifying the metric and preference for offline WAN connections 56

IPX features 58

SPX spoofing added for IPX 58
New Answer profile option for dial-in NetWare clients 58
Support for IPX without defining an IPX server 59

SNMP features 60

Enable and Disable individual modem using SNMP 60
SNMP write security disabled by default 60
SNMP "get" now retrieves MPP session statistics 61
SNMP can monitor WAN lines and channels 64
SNMP can obtain active call status 68
SNMP system reset 71
SNMP can detect concurrent sessions 71
SNMP RFC 1398 Ethernet-like MIB (dot3) support added 73
Set system clock through SNMP 74
Terminating user sessions using SNMP 74
Ascend MIB change supports counters for total and current calls 75
Firewall Control Protocol managed by SNMP 77
SNMP request authentication added 80
SNMP sysConfigTftpStatus object reports more states 81
SNMP now reports reasons for last reset 82

Tunneling features 84

Maximum number of ATMP Tunnel sessions can be set 84
ATMP inactivity timer 85

Administration features 85

Data rates reported to syslog 86
Receive and transmit rates now reported 86
Syslog enhancements 87
Syslog messages generated when Security profile activated 89
Configure port for Syslog messages 90
Defender authentication enhancements 91
RLOGIN enhanced -l option 92
TFTP checks for compatibility of downloaded files 93

Appletalk features 94

AppleTalk routing added 94
Defender authentication added for AppleTalk Remote Access Protocol (ARAP) 103
Dial-in PPP support for AppleTalk 104
SecurID authentication for AppleTalk Remote Access (ARA) users 106

Modem features 107

Megahertz CC1336 and XJ1336 modems available with MAX 200Plus 107
USR Courier V.Everything modem available with MAX 200Plus 108
Support for Xircom 33.6 PC Card modems 109
Support for Hayes Optima 33.6 PC Card (PCMCIA) modems 110
New modems supported on the MAX 200Plus 111
Support for Practical Peripherals 33.6 V.34+Fax PC Card modems 112
Support for Practical Peripherals 33.6 PC Card modem with EZ-Port 113
Support for the Hayes Optima 33.6 PC Card modem with EZjack 114
Support for Hayes Optima 288 V.34+Fax PC Card modems (Australia) 115
Flashing CD lights indicate problems with modems 117

Introduction

This addendum applies to the MAX 200Plus.

What is in this addendum

The documentation that came with your MAX unit describes how to install the hardware and configure the system. However, since the documentation was published, new system software has been released that contains features that are not yet included in the product documentation. This addendum describes those new features.

Related publications

Additional information is available in the MAX documentation set. The MAX documentation set consists of the following manuals:

- *MAX 200Plus Getting Started Guide.* Explains how to install the MAX hardware. Includes the MAX technical specifications.
- *MAX 200Plus ISP & Telecommuting Configuration Guide*. Explains how to use the VT100 interface to configure WAN connections and other related features.
- *MAX 200Plus Reference Guide.* An alphabetic reference to all MAX profiles, parameters, and commands.
- *MAX 200Plus Security Supplement.* Explains how to configure the MAX built-in security features. For information about configuring Secure Access Firewalls, see the documentation that came with your software.
- *MAX 200Plus Windows 95 Client Configuration Guide.*
- *MAX 200Plus Administrator's Guide.* Explains how to use the Window GUI interface to configure WAN connections and other related features.

A few words about this release...

Version 6.0.0 is the first product of a new software release process, which applies expanded internal testing designed to detect as many problems as possible prior to release. The new process adds layers of testing to Ascend's previous test suite.

We have endeavored to resolve any problems that might have major negative effects for your network, and to verify that the problems have been fixed. Some Trouble Reports (TRs) are still open, but this release resolves many of the highest priority issues. To fully complete our release process, we have added a formal beta test program, which you are invited to join for future releases, at www.ascend.com. This program and our newly expanded internal testing suite help ensure that each general release is tested as thoroughly as possible before we formally introduce it.

Our new software release process simplifies and unifies the version numbering we use for all products and integrates the MAX and Pipeline product lines into a single release.

Note: This release also introduces new software upgrade procedures, so please see the upgrade instructions in "Upgrading system software" on page 3.

A new section of the Release Notes, *Known issues*, describes open issues that might affect your environment. Some issues affect functionality. Others cause no functional problems, but can affect the VT100 display or terminal-server screen.

We know you'll see benefits from our new release process and renewed dedication to the quality of our releases.

Thank you,

Larry Gray
Director, Software Quality Assurance

Dana Harrison
Product Line Director, MAX products

Upgrading system software

Caution: The procedure for uploading new software to Ascend units have changed significantly. Carefully read the new software loading procedures explained in this section before upgrading your system.

This section explains how to upgrade your system software. It contains the following sections:

- Definitions and terms
- Guidelines for upgrading system software
- Before you begin
- Upgrading system software with a standard load
- Upgrading system software with a fat or thin load
- Upgrading system software with an extended load
- Upgrading system software from versions earlier than 4.6C to version 5.0A or above
- Using the serial port to upgrade to a standard or a thin load
- System messages

Definitions and terms

This document uses the following terms:

Build	The name of the software binary. For example, `ti.m40` is the MAX 4000 T1 IP-only software build. For the names of all the software builds and the features they provide see `/pub/Software-Releases/Max/Upgrade-Filenames.txt` or `/pub/Software-Releases/Pipeline/Upgrade-Filenames.txt` on the Ascend FTP server. If possible, you should stay with the same build when upgrading. Loading a different build can cause your Ascend unit to lose its all or part of its configuration. If this happens, you must restore your configuration from a backup.
Standard load	Software versions 4.6Ci18 or earlier and all 4.6Cp releases. You can load these versions of software through the serial port or by using TFTP. TFTP is the recommended upgrade method for standard loads.
Fat load	4.6Ci19 to 5.0Aix and all 5.0Ap releases with a file size greater than 960 KB (for MAX units) or 448K (for Pipeline units). Before upgrading to a fat load for the first time, you must upgrade to a thin load. You must use TFTP to upgrade to fat loads.

Thin load	4.6Ci19 to 5.0Aix and all 5.0Ap releases with a file size less than 960 KB (for MAX units) or 448 KB (for Pipeline units). TFTP is the recommended upgrade method for thin loads.
Restricted load	6.0.0 or later MAX release denoted by an “r” preceding the build name. For example, rti.m40 is the restricted load for the MAX 4000 T1 IP-only software build. Before upgrading to an extended load for the first time, you must upgrade to a restricted load. A restricted load only contains essential system software and is not meant to be run in a working environment. It does not have full functionality and is to be used only to upload to an extended load. TFTP is the recommended upgrade method for restricted loads. Pipeline releases do not have restricted loads.
Extended load	6.0.0 or later MAX release denoted by an “f” preceding the build name. You must use TFTP to upgrade to extended loads. For example, `fti.m40` is the extended load for the MAX 4000 T1 IP-only software build. Pipeline releases do not have extended loads.

Guidelines for upgrading system software

Caution: Before upgrading, consider the following very important guidelines:

- Use TFTP to upgrade if possible. TFTP is more reliable and saves the Ascend unit configuration when you upgrade.
- You cannot load a fat load or an extended load through the serial port. You must use TFTP.
- If you are using TFTP to upgrade your software, use the `fsave` command immediately after executing the `tload` command. Failure to do so might cause your Ascend unit to lose its configuration.
- If possible, you should always stay with the same build of software when you upgrade. If you load a different version, your Ascend unit may lose its configuration. If this happens, you must restore your configuration from a backup.
- If you are upgrading to a software version 5.0A or 5.0Aix fat load for the first time, you must be on a load that supports the fat load format. All versions of software 5.0A or above support fat loads. You should perform the upgrade in two steps:
  - Upgrade to a thin load of the same build
  - Upgrade to the fat load
- If you are upgrading to a software version 6.0.0 or above, you must be on a load that supports the extended load format. All versions of software 6.0.0 or above support extended loads. You should perform the upgrade in two steps:
  - Upgrade to a restricted load of the same build
  - Upgrade to the extended load
- You can upgrade to a thin load or a restricted load from any version of software.

- If you are upgrading from software version 4.6C or earlier to software version 5.0A or later, see "Upgrading system software from versions earlier than 4.6C to version 5.0A or above" on page 11 for important information before you start.

Table 1 explains where to find the information you need to upgrade your unit.

Table 1. Ascend system software versions

Version you are upgrading to	Use the instructions in...
Standard load (4.6Ci18 or earlier and all 4.6Cp releases)	"Upgrading system software with a standard load" on page 6.
Fat or thin load (4.6Ci19 to 5.0Aix and all 5.0Ap releases)	"Upgrading system software with a fat or thin load" on page 7.
Extended load (6.0.0 or later)	A restricted load only contains essential system software and is not meant to be run in a working environment. It does not have full functionality and is to be used only to upload to an extended load. "Upgrading system software with an extended load" on page 10.

Before you begin

Make sure you perform all the tasks explained in Table 2 before upgrading your software.

Table 2. Before upgrading

Task	Description
If necessary, activate a Security Profile that allows for field upgrade.	If you are not sure how, see the section about Security Profiles in your documentation.
Record all of the passwords you want to retain, and save your Ascend unit's current configuration to your computer's hard disk.	For security reasons, passwords are not written to configuration files created through the serial console. A configuration file created using the Tsave command, however, *does* contain the system passwords. You can restore the Tsave configuration file using the serial console. If you chose to save your configuration using the serial console, you will have to restore your passwords manually. Restoring passwords is explained in "Using the serial port to upgrade to a standard or a thin load" on page 12.

Table 2. Before upgrading (continued)

Task	Description
Obtain the correct file, either by downloading it from the FTP server or by requesting it from Ascend technical support.	To ensure that you load the correct software binary, you should check the load currently installed on your unit. To do so: **1** Tab over to the 00-100 Sys Options window. **2** Press Enter to open the Sys Options menu. **3** Using the Down-Arrow key (or Ctrl- N), scroll down until you see a line similar to the following: `Load: tb.m40` **4** When upgrading, obtain the file with same name from the Ascend FTP site. If your unit does not display the current load or you are unsure about which load to use, contact technical support.
If you are upgrading to a fat load or an extended load for the first time, you must also obtain a thin load or a restricted load of the same build, if possible.	For example, if you are upgrading a MAX 4000 to 5.0Ai13 fat load (such as `tbim.m40`), obtain a thin load of the same build (such as 5.0A `tbim.m40`). If you are upgrading to a MAX 6.0.0 extended load, obtain a 6.0.0 restricted load. Restricted loads are designated with an "r" in the load name. (For example `rtbam.m40` is a restricted load). Newer Pipeline 50 or 75 units do not have fat loads and no Pipeline units have extended or restricted loads. Refer to `/pub/Software-Releases/Pipeline/Upgrade-Filenames.txt` to determine if you have a new Pipeline 50 or 75 unit.
If you are using TFTP, make sure you load the correct binaries into the TFTP home directory on the TFTP server.	You must use TFTP to upgrade to a fat load or an extended load.
If you are using the serial port, make sure you have a reliable terminal emulation program, such as Procomm Plus.	If you use the serial port, you can only upgrade to a standard or a thin load. Upgrading through the serial port is not recommended. If you use a Windows-based terminal emulator such as Windows Terminal or HyperTerminal, disable any screen savers or other programs or applications that could interrupt the file transfer. Failure to do so might cause the software upload to halt, and can render the Ascend unit unusable.

Upgrading system software with a standard load

To upgrade system software with a standard load you can use either the serial port or TFTP. TFTP is the recommended method because it preserves your Ascend unit's configuration. If you want to use the serial port to upgrade, see "Using the serial port to upgrade to a standard or a thin load" on page 12.

Using TFTP to upgrade to a standard load

To upgrade to a standard load using TFTP, you only have to enter a few commands. But you must enter them in the correct sequence, or you could lose the Ascend unit's configuration.

To upgrade to a standard load via TFTP:

1. Obtain the software version you want to upgrade to and place it in the TFTP server home directory.
2. From the Ascend unit's VT100 interface, access the diagnostics monitor by typing the following characters in rapid succession:

   Esc [Esc =

   Or, press Ctrl-D to invoke the DO menu and select D=Diagnostics.
3. At the > prompt, use the Tsave command to save your configuration as in the following example:

   > **`tsave tftp-server router1.cfg`**

   This saves the configuration of your unit to the file named `router1.cfg` in the TFTP home directory of the server named `tftp-server`. This file must already exist and be writable. Normally, TFTP upgrades save the configuration. Tsave is a precaution.

Caution: The file you save with the Tsave command contains all the passwords in clear text. You should move this file from the TFTP directory to a secure location after the upgrade procedure is complete.

4. Enter the following command:

   `tloadcode hostname` **`filename`**

   where *hostname* is the name or IP address of your TFTP server, and *filename* is the name of the system software on the server (relative to the TFTP home directory).

   For example, the command:

   `tloadcode tftp-server t.m40`

   loads `t.m40` into flash from the machine named `tftp-server`.

Caution: You must use the Fsave command immediately after executing the Tload command. Failure to do so can cause your Ascend unit to lose its configuration.

5. Enter the following command to save your configuration to flash memory:

   `fsave`
6. Enter the following command:

   `nvramclear`

   After the Ascend unit clears NVRAM memory, it automatically resets.

This completes the upgrade.

Upgrading system software with a fat or thin load

Upgrading to a fat or thin load is not difficult, but you must be careful to follow the correct sequence of tasks.

Caution: If you are upgrading from software version 4.6C or earlier, see "Upgrading system software from versions earlier than 4.6C to version 5.0A or above" on page 11 for important information before upgrading.

To upgrade your system:

1 Obtain the software version binary you want to upgrade to and place it in the TFTP server home directory. If you are upgrading to a fat load for the first time, also obtain a thin load of the same build and place it in the same directory. (See page "Definitions and terms" on page 3 for an explanation of fat and thin loads.)

Caution: If possible, you should stay with the same build when upgrading. Loading a different build can cause your Ascend unit to lose all or part of its configuration. If this happens, you must restore your configuration from a backup.

For example, if you are upgrading a MAX 4000 to 5.0Ai13 fat load (such as `tbim.m40`), obtain a thin load of the same build (such as 5.0A `tbim.m40`).

Note: Newer Pipeline 50 or 75 units do not have fat or thin loads, you only need to load a single software binary. Refer to `/pub/Software-Releases/Pipeline/Upgrade-Filenames.txt` on the Ascend FTP site to determine if you have a new Pipeline 50 or 75 unit.

2 From the Ascend unit's VT100 interface, access the diagnostics monitor by typing the following characters in rapid succession:

Esc [Esc =

Or, press Ctrl-D to invoke the DO menu and select D=Diagnostics.

3 At the **>** prompt, use the Tsave command to save your configuration, as in the following example:

> **`tsave tftp-server router1.cfg`**

This saves the configuration of your unit to the file named `router1.cfg` in the TFTP home directory of the server named `tftp-server`. This file must already exist and be writable. Normally, TFTP upgrades save the configuration. Tsave is a precaution.

Caution: The file you save with the Tsave command contains all the passwords in clear text. You should move this file from the TFTP directory to a secure location after the upgrade procedure is complete.

4 At the **>** prompt, enter:

> **`tloadcode`** ***`hostname filename`***

where *hostname* is the name or IP address of your TFTP server, and *filename* is the name of the system software on the server (relative to the TFTP home directory).

Caution: If you are upgrading from a standard load to a fat load, make sure you load a thin load first.

For example, the command:

`> tloadcode tftp-server t.m40`

loads `t.m40` into flash from the machine named `tftp-server`.

Caution: You must use the Fsave command immediately after executing the Tload command. Failure to do so may cause your Ascend unit to lose its configuration.

5 Enter the following command to save your configuration to flash memory:

```
fsave
```

6 Enter the following command:

```
nvramclear
```

After the Ascend unit clears NVRAM memory, it automatically resets.

7 If you are upgrading to a thin load, you are done. If you are upgrading to a fat load, repeat the procedure, this time uploading the fat load binary.

After a successful upgrade, one of the following messages appears.

- If the load is thin:

```
UART initialized
thin load: inflate
.....................................................
starting system...
```

- If the load is fat:

```
UART initialized
fat load: inflate
.....................................................
starting system...
```

This completes the upgrade if you have no errors. If the upgrade is not successful, refer to ”Recovering from a failed fat load upgrade” next.

Recovering from a failed fat load upgrade

If a fat load has a CRC (cyclic redundancy check) error, the following message appears:

```
UART initialized
fat load: bad CRC!!
forcing serial download at 57600 bps
please download a "thin" system...
```

Immediately after this message appears, the serial console speed is switched to 57600 bps, and the Ascend unit initiates an Xmodem serial download. To recover from this error and load the fat system, you must first load a thin system that is fat-load aware. Proceed as follows:

1 Activate your Xmodem software.

2 After you have finished loading the fat-aware thin load, reboot the unit.

3 Use the Tload command to download the fat load.

When you download a fat load, messages similar to the following appear on the diagnostics monitor screen:

```
> tload 192.168.1.82 tbam.m40
saving config to flash
.....................................
loading code from 192.168.1.82:69
file tbam.m40..
fat load part 1:
.............................................................
```

```
......
fat load part 2:
.................................................
```

The "fat load part *n*:" messages notify you when the first and second halves of the download begin.

Upgrading system software with an extended load

Your first upgrade to an extended load requires a preliminary procedure. You must first upgrade to a restricted load. A restricted load only contains essential system software and is not meant to be run in a working environment. It does not have full functionality and is to be used only to upload to an extended load. Note that Pipeline units do not have extended loads.

Warning: You cannot upgrade to extended loads using an IP over X.25 connection because restricted loads do not have X.25 support.

Caution: If you are upgrading from software version 4.6C or earlier, see "Upgrading system software from versions earlier than 4.6C to version 5.0A or above" on page 11 for important information before upgrading.

To upgrade your system:

1. Obtain the software-version binary you want to upgrade to and place it in the TFTP server home directory.

   Extended loads are denoted by an "f" preceding the build filename.

2. If this is the first time you have upgraded to an extended load, obtain a restricted load of the same build and place it in the directory.

   For example, if you are upgrading a MAX 4000 to an extended load (such as `tbam.m40`), obtain a MAX 4000 restricted load (such as `rtbam.m40`).

3. From the Ascend unit's VT100 interface, access the diagnostics monitor by typing the following characters in rapid succession:

   Esc [Esc =

   Or, press Ctrl-D to invoke the DO menu, and select D=Diagnostics.

4. At the > prompt, use the Tsave command to save your configuration, as in the following example:

   > **`tsave tftp-server router1.cfg`**

   This saves the configuration of your unit to the file named `router1.cfg` in the TFTP home directory of the server named `tftp-server`. This file must already exist and be writable. Normally, TFTP upgrades save the configuration. Tsave is a precaution.

Caution: The file you save with the Tsave command contains all the passwords in clear text. You should move this file from the TFTP directory to a secure location after the upgrade procedure is complete.

5. At the > prompt, enter:

   `tloadcode` ***`hostname filename`***

   where *hostname* is the name or IP address of your TFTP server, and *filename* is the name of the system software on the server (relative to the TFTP home directory).

Caution: If you want to upgrade your system for the first time to a software version 6.0.0 or later, you must first upgrade your system to a restricted load. Failure to do so can cause your Ascend unit to lose its configuration.

For example, the command:

tloadcode tftp-server rtbam.m40

loads the restricted load `rtbam.m40` into flash from the machine named `tftp-server`.

Caution: You must use the Fsave command immediately after executing the Tload command. Failure to do so can cause your Ascend unit to lose its configuration.

6 Enter the following command to save your configuration to flash memory:
`fsave`

7 Enter the following command:
`nvramclear`
After the Ascend unit clears NVRAM memory, it automatically resets.

If you have downloaded the extended load, the upgrade is complete.

If you have loaded a restricted load, your system boots up in restricted mode. Restricted mode only allows you to load software. You cannot change or save profiles. While in restricted mode, the Edit menu displays the following banner:

* * RESTRICTED MODE * * * YOU MUST RERUN THE LAST tloadcode COMMAND *

If your system boots up in restricted mode, perform the following steps:

1 At the > prompt, enter:
`tloadcode` ***`hostname filename`***
where *hostname* is the name or IP address of your TFTP server, and *filename* is the name of the extended load of system software on the server (relative to the TFTP home directory).
For example, the command:

`tloadcode tftp-server ftbam.m40`

loads the extended load `ftbam.m40` into flash from the machine named `tftp-server`.

2 Enter the following command:
`nvramclear`
After the Ascend unit clears NVRAM memory, it automatically resets.

Your system will then boot up with the new version of software running.

Upgrading system software from versions earlier than 4.6C to version 5.0A or above

If you are upgrading from software version 4.6C or earlier to version 5.0A or later, perform the upgrade in the following order:

1. Load version 4.6Ci18, following the procedure in "Upgrading system software with a standard load" on page 6.
2. Load version 5.0A, following the procedure in "Upgrading system software with a fat or thin load" on page 7.
3. Load version 5.0Aix or 6.0.0, following the procedure in "Upgrading system software with a fat or thin load" on page 7 (for software versions 5.0Aix) or "Upgrading system software with an extended load" on page 10 (for software version 6.0.0).

Caution: Failure to follow this procedure might cause your Ascend unit to lose or corrupt its configuration, and could render the unit unusable.

Using the serial port to upgrade to a standard or a thin load

Caution: Uploading system software via the serial console overwrites all existing profiles. Save your current profiles settings to your hard disk before you begin upgrading system software. After the upgrade, restore your profiles from the backup file you created. Since the backup file is readable text, you can reenter the settings through the Ascend unit's user interface. To avoid having existing profiles overwritten, use TFTP to upgrade your unit.

Caution: You cannot upload a fat load or an extended load using the serial port; it must be done using TFTP.

Upgrading through the serial port consists of the following general steps:

- Saving your configuration
- Uploading the software
- Restoring the configuration

Before you begin

Before upgrading your system through the serial port, make sure you have the following equipment and software:

- An IBM compatible PC or Macintosh with a serial port capable of connecting to the Ascend unit's Console port.
- A straight-through serial cable.
- Data communications software for your PC or Mac with XModem CRC/1K support (for example, Procomm Plus, HyperTerminal for PCs or ZTerm for the Mac).

Caution: If you use a Windows-based terminal emulator such as Windows Terminal or HyperTerminal, disable any screen savers or other programs or applications that could interrupt the file transfer. Failure to do so might cause the software upload to halt, and can render the Ascend unit unusable.

Saving your configuration

Before you start, verify that your terminal emulation program has a disk capture feature. Disk capture allows your emulator to capture to disk the ASCII characters it receives at its serial

port. You should also verify that the data rate of your terminal emulation program is set to the same rate as the Term Rate parameter in the System Profile (Sys Config menu).

You can cancel the backup process at any time by pressing Ctrl-C.

To save the Pipeline configuration (except passwords) to disk:

1 Open the Sys Diag menu.

2 Select Save Config, and press Enter.

The following message appears:

Ready to download - type any key to start....

3 Turn on the Capture feature of your communications program, and supply a filename for the saved profiles. (Consult the documentation for your communications program if you have any questions about how to turn on the Capture feature.)

4 Press any key to start saving your configured profiles.

Rows of configuration information appear on the screen as the configuration file is downloaded to your hard disk. When the file has been saved, your communications program displays a message indicating the download is complete.

5 Turn off the Capture feature of your communications program.

6 Print a copy of your configured profiles for later reference.

You should examine the saved configuration file. Notice that some of the lines begin with `START=` and other lines begin with `END=`. A pair of these START/STOP lines and the block of data between them constitute a profile. If a parameter in a profile is set to its default value, it does not appear. In fact, you can have profiles with all parameters at their defaults, in which case the corresponding START/STOP blocks are empty. Make sure that there are no extra lines of text or characters either before `START=` or after `END=`. If there are, delete them. They could cause problems when you try to upload the file to the Ascend unit.

Uploading the software

To upload the software:

1 Type the following four-key sequence in rapid succession (press each key in the sequence shown, one after the other, as quickly as possible):

Esc [Esc -

(Press the escape key, the left bracket key, the escape key, and the minus key, in that order, in rapid succession.) The following string of Xmodem control characters appears:

CKCKCKCK

If you do not see these characters, you probably did not press the four-key sequence quickly enough. Try again. Most people use both hands and keep one finger on the escape key.

2 Use the Xmodem file-transfer protocol to send the system file to the Ascend unit.

Your communications program normally takes anywhere from 5 to 15 minutes to send the file to your Ascend unit. The time displayed on the screen does not represent real time. Do not worry if your communication program displays several “bad batch” messages. This is normal.

After the upload, the Ascend unit resets. Upon completion of the self-test, the Ascend unit’s initial menu appears in the Edit window with all parameters set to default values. This completes the upgrade.

If the upload fails during the transfer, try downloading another copy of the binary image from the Ascend FTP server and re-loading the code to the Ascend unit. If you still have problems, contact Ascend technical support for assistance.

Restoring the configuration

Under certain circumstances, the serial-port method might not completely restore your configuration. You should therefore verify that your configuration was properly restored every time you use this method. If you have many profiles and passwords, you should consider using TFTP to upgrade your software. (See "Using TFTP to upgrade to a standard load" on page 7.)

To restore the configuration, you must have administrative privileges that include Field Service (such as the Full Access Profile, for example). You use the Restore Cfg command to restore a full configuration that you saved by using the Save Cfg command, or to upload more specific configuration information obtained from Ascend (for example, a single filter stored in a special configuration file).

To load configuration information through the serial port

1. From the Ascend unit's VT100 interface, access the diagnostics monitor by typing the following characters in rapid succession:

   Esc [Esc =

   Or, press Ctrl-D to invoke the DO menu, and select D=Diagnostics.
2. At the **>** prompt, enter the Fclear command:

   > **`fclear`**
3. At the **>** prompt, enter the NVRAMClear command:

   > **`nvramclear`**

   This causes the system to reset. When it comes back up, proceed with restoring your configuration.
4. Enter **`quit`** to exit the Diagnostic interface.
5. Open the Sys Diag menu.
6. Select Restore Cfg, and press Enter.

   The following message appears:

   ```
   Waiting for upload data...
   ```
7. Use the Send ASCII File feature of the communications software to send the configuration file to the unit. (If you have any questions about how to send an ASCII file, consult the documentation for your communications program.)

   When the restore has been completed, the following message appears:

   ```
   Restore complete - type any key to return to menu
   ```
8. Press any key to return to the configuration menus.
9. Reset the Ascend unit, by selecting System > Sys Diag > Sys Reset and confirming the reset.

Restoring passwords

For security reasons, passwords are not written to configuration files created through the serial console. A configuration file created using the Tsave command, however, *does* contain the system passwords. You can restore the Tsave configuration file using the serial console.

After upgrading you may have to re-enter all the passwords on your system. If you edit your saved configuration file, however, and enter passwords in the appropriate fields (by replacing the word *SECURE* in each instance), these passwords will be restored. But note that if you do choose to edit your configuration file, you must save it as text only or you will not be able to load it into your unit.

If you restored a complete configuration, the passwords used in your Security profiles have been wiped out. To reset them:

1. Press Ctrl-D to invoke the DO menu, select Password, and choose the Full Access profile.
2. When you are prompted to enter the password, press Enter (the null password).

   After you have restored your privileges by entering the null password, you should immediately open the Connection profiles, Security profiles, and Ethernet profile (Mod Config menu), and reset the passwords to their previous values.

System messages

Table 3 explains the messages that can appear during your upgrade.

Table 3. System software messages

Message	Explanation

`UART initialized`
`fat load: bad CRC!!`
`forcing serial download at 57600 bps`
`please download a "thin" system...`
	The fat load has a CRC (cyclic redundancy check) error. Immediately after this message appears, the serial console speed is switched to 57600 bps, and the Ascend unit initiates an Xmodem serial download. Load a thin load that understand the fat load format, as explained in “Upgrading system software with a fat or thin load” on page 7.

`File tbam.m40`
`incompatible fat load format--`
`discarding downloaded data`
	You attempted to upgrade to a fat load from a version of system software that does not understand the fat load format. You must first load a thin load that is fat load aware, as explained in “Upgrading system software with a fat or thin load” on page 7.

`This load has no platform identifier. Proceed with caution.`	This message can occur if you are running software version 5.0Ai11 or later and you load an earlier incremental or patch release onto your system. The message indicates that Tloadcode cannot determine which platform the code is intended for. If you are using the correct software version, you can ignore this message.
`This load appears not to support your network interface.` `Download aborted. Use ‘tloadcode -f’ to force.`	Indicates you are attempting to load a version of code intended for a different network interface (for example, loading MAX 4000 T1 software onto a MAX 4000 E1 unit).

Table 3. System software messages (continued)

Message	Explanation
`This load appears to be for another platform.` `Download aborted. Use 'tloadcode -f' to force.`	Indicates you are attempting to load a version of code onto a platform for which it is not intended (for example, loading MAX 4000 software onto a MAX 2000). This is not recommended

`UART initialized`
`fat load: inflate`
`.....`
`.....`
`starting system...`
	Indicates you have successfully loaded a fat load.

`UART initialized`
`extended load:`
`inflate essential`
`.+.+.....`
`invalid CRC!!`
`entering restricted mode`
`starting system...`
	Indicates the extended load has failed and that your system is being brought up in restricted mode. You must reload the software as explained in "Upgrading system software with an extended load" on page 10.

`UART initialized`
`extended load:`
`inflate essential`
`.+.+.....`
`.`
`invalid length!!`
`entering restricted mode`
`starting system...`
	Indicates the extended load has failed and that your system is being brought up in restricted mode. You must reload the software as explained in "Upgrading system software with an extended load" on page 10.

UART initialized extended load: inflate essential .+.+..... inflate expendable.....|.... `starting system...`	Indicates you have successfully loaded an extended load.

`UART initialized`
`thin load: inflate`
`.....`
`.....`
`starting system...`
	Indicates you have successfully loaded a thin load.

Known issues

Known issues with this software release affect IPX and the display. There are also a couple of other issues, which will be resolved shortly.

IPX issues

Four known issues involve IPX. The first is that NetWare, by default, relies on the Data Link layer (also called Layer 2) to validate and guarantee data integrity. STAC link compression, if used, generates an eight-bit checksum, which is inadequate for NetWare data. This issue has existed in all previous releases of the software.

If your MAX supports NetWare (either routed or bridged), and you require link compression, you should configure your MAX in one of the following ways:

- Configure either STAC-9 or MS-STAC link compression, which use a more robust error-checking method, for any connection profile supporting IPX data. Configure link compression in the Ethernet > Answer > PPP Options > Link Comp parameter and Ethernet > Connections > *Any Connection profile* > Encaps Options > Link Comp parameter.
- Enable IPX-checksums on your NetWare servers and clients. (Both server and client must support IPX-checksums. If you enable checksums on your servers but your clients do not support checksums, they will fail to log in successfully.)
- Disable link compression completely by setting Ethernet > Answer > PPP Options > Link Comp = None and Ethernet > Connections > *Any Connection profile* > Encaps Options > Link Comp = None. By disabling link compression, the MAX validates and guarantees data integrity by means of PPP.

Following are the other IPX issues:

- When you bring up a call with the IPXPING command, *Answer* appears in the IPX Route Table as an entry under the *Origin* column. When the call is disconnected, the table returns to normal. This issue does not affect IPX functionality.
- The MAX does not send IPX RIP Response packets for a PPP-encapsulated dialup connection.
- If you support ATMP tunnels for your IPX clients, do *not* upgrade to 6.0.0.

Modem out-dial issues

The following issues can affect modem out dial:

- When dialing a busy number, terminal-server out-dial modems do not report BUSY. The MAX detects the busy signal and goes on-hook before its modem receives the busy signal. The MAX reports Cause Code 17 in the message log, the modem disconnects, and the screen displays NO CARRIER. This might be a problem for users or scripts that require a modem to report BUSY.

Display-only issues

The following issues cause display symptoms, but no functional problems:

- Occasionally, when you enter the terminal-server command `Open`, the MAX displays random characters on the screen.
- The MAX displays the Ethernet > Filters > NetWare Call filter for all loads, regardless of whether or not they support IPX.

Other issues

The following issues will be resolved shortly in a maintenance release:

- When Stacking is enabled, PAP-TOKEN-CHAP is not supported.
- Data filters are not supported when used in conjunction with Microsoft's CallBack Control Protocol (CBCP)
- Windows 95 machines installed with Winsock.dll (version 2.0) fail to log in successfully. You should configure Windows 95 in one of the following ways:
  - Upgrade Windows 95 with the latest Microsoft Dialup Networking patch 1.2, msdun12.exe.
  - Revert to older versions of Winsock.dll.

WAN access features

MAX CH CNT upper limit specific to platform

Now, the Pipeline or MAX will only allow you to configure Max Ch Cnt to a supported range of values. For example, the Max Ch Cnt parameter on a P50 can only be set to a maximum of 2. On the MAX 200+, it can be set to 8. On the MAX 4000, it can be set to 32.

Previously, the Max Ch Cnt parameter in each Connection Profile for every Pipeline or MAX unit was allowed any value from 0 to 32, regardless of whether the device supported 32 channels or not.

B-Channel preference when MP+ or BACP adds bandwidth

Previously, when an additional channel was requested due to an increase in traffic, the MAX randomly selected a B-channel for use. WIth this release, BACP and MP+ direct the MAX to give preference to the second B-channel of the BRI line on which the call originated.

Some Internet Service Providers offer discounts if multi-channel sessions use the same BRI line instead of spanning BRI lines.

BACP/BAP PPP protocol IDs match IETF

The protocol IDs for BACP/BAP PPP on Ascend units have been changed to conform to the protocol IDs specified by the IETF. These changes are internal and cannot be modified by the user; there are no changes to the user interface.

Note: Both sides of a connection must use the same version of the BACP/BAP protocol IDs in order to connect successfully.

The protocol IDs that have changed are as follows:

Table 4.

Previous Protocol ID	Current Protocol ID
8071	c02b.
0071	c02d

TACACS+ server retry attempts

TACACS+ backup server

This note adds the information to the documentation that the server makes two attempts to connect to each server.

How it works

The MAX sends a request for authentication to the first server on the list of hosts specified by Auth Host and waits for a response from the server for the number of seconds specified in the Auth Timeout parameter. If the MAX does not receive a response, within that time, it sends a second request for authentication to the same server and waits for the same amount of time. If the MAX does not receive a response within the specified timeout, it sends a request for authentication to the next server on the Auth Host list and repeats the process.

If the MAX is unsuccessful in obtaining a response from any of the servers on the list, the connection fails.

Set Cause Code for ISDN DISCONNECT

Overview

You can now specify that either User busy or Normal call clearing is sent as the Cause Element value in ISDN DISCONNECT packets when either Calling ID or Called ID Authentication fails.

There are two possible reasons the MAX sends out a DISCONNECT message with respect to ID authentication:

- due to a mismatch between the actual number and the expected number
- due to a RADIUS timeout

With this release, if either Calling ID or Caller ID authentication fails, the MAX can send out either of two Cause codes in the DISCONNECT message:

- User Busy
- Normal Call Clearing

The names and behaviors of two parameters in the Auth submenu of the Ethernet profile have been modified to support this feature. You can specify which Cause code the MAX sends out in DISCONNECT packets in these parameters:

- CLID Timeout Busy is now Timeout Busy
- CLID Fail Busy is now ID Fail Busy

Parameter Reference

Timeout Busy (previously CLID Timeout Busy)

Description: Specifies whether to return User Busy or Normal Call Clearing as a Cause in IDSN DISCONNECT messages when ID authentication fails due to a RADIUS timeout.

Usage: Press Enter to toggle between Yes and No. No is the default. If you choose Yes, and the ID authentication fails due to a RADIUS timeout, the DISCONNECT message will have the Cause value User Busy (decimal value 17). If you choose No, the Cause value will be Normal Call Clearing (decimal value 16).

Dependencies: This parameter will be N/A if Auth=None or Auth=TACACS+ in the this profile. The value set in this parameter applies to both Caller ID and Called ID authentication.

This parameter is N/A if ID Auth=Ignore.

Location: Ethernet Profile: Ethernet > Mod Config > Auth

See Also: IDFail Busy,

ID Fail Busy (previously CLID Fail Busy)

Description: Specifies whether to return User Busy or Normal Call Clearing as a Cause in IDSN DISCONNECT messages when authentication fails due to a mismatch between the actual number and the expected number.

Usage: Press Enter to toggle between Yes and No. No is the default. If you choose Yes, and the ID authentication fails due to a mismatch between the actual number and the expected number, the DISCONNECT message will have the Cause value User Busy (decimal value 17). If you choose No, the Cause value will be Normal Call Clearing (decimal value 16).

Dependencies: This parameter will be N/A if Auth=None or Auth=TACACS+ in the this profile. The value set in this parameter applies to both Caller ID and Called ID authentication.

This parameter is N/A if ID Auth=Ignore.

Location: Ethernet Profile: Ethernet > Mod Config > Auth

See Also: Timeout Busy,

Changes to Ascend MIB

Two values have been added to possible values for eventDisconnectReason in the Ascend MIB:

- clidAuthFailed (value 4)

  This value indicates that the reason for the failure to authenticate is a mismatch between the expected value and the value provided by the caller.
- clidAuthServTimeout (value 5)

  This value indicates that the reason for the failure to authenticate is a server timeout.

The Ascend MIB is updated frequently. You should always use the most current MIB. You can download the latest version of the Ascend MIB from the Ascend FTP server.

Raw TCP connection enabled

Users of the MAX's terminal server in the menu mode get a list of hosts to which they can telnet. With this software, that list can now configure include hosts to which the terminal server users can make a raw TCP connection.

Overview

In some instances, a terminal server user needs a raw TCP connection instead of a Telnet connection. This feature allows you to include the IP addresses and/or DNS names of a hosts to which the user can make a raw TCP connections.

Note: You cannot configure raw TCP hosts if you are using a RADIUS server to provide the list of hosts.

The IP address field for the TServ Options menu now holds a text string of up to 31 characters, instead of the dotted-decimal IP address format previously required.

Upgrading to a system software version containing this feature

To upgrade to a software version containing this feature, you must use nvramclear. This is required because the size of a field has changed. See the installation instructions that accompany new system downloads on the Ascend FTP server for more information on installing new system software.

Configuring a raw TCP host

1. Open the Ethernet > Mod Config > TServ options menu.
2. Select one of the Host # Addr fields and enter the following:

rawTcp *hostaddress portnumber*

rawTCP is the required string that causes the MAX to establish a raw TCP connection when the user chooses this host number. This entry is case-sensitive and must be entered exactly as shown.

hostname can be the DNS name of the host or the IP address of the host. The total number of characters, including the rawTcp string, must not exceed 31.

portnumber is the number of the port on which the connection for this host is to be established.

3 Enter a description of the host on the Host # Text field.

How it works

For example, assume the following entries in the TServ Options menu:

```
Remote Conf=No
Host #1 Addr=137.175.2.11
Host #1 Text=v
Host #2 Addr=
Host #2 Text=
Host #3 Addr=
Host #3 Text=
Host #4 Addr=rawTcp v 7
Host #4 Text=v 7
Immed Service=None
Immed Host=N/A
Immed Port=N/A
Telnet Host Auth=No
```

The Terminal Server menu would contain something like the following:

```
** Ascend Pipeline Terminal Server **

     1. v
     2. v 7

     Enter Selection (1-2,q)
```

If a user picks host #4, a raw TCP connection is established to the host "v" on port 7.

If a user picks host #1, a Telnet connection is established to the host 137.175.2.11. In this case, the port is the default Telnet port.

Host #*n* Addr

Description: The Host #n Addr parameter has been modified to accept a string in the format of

rawTcp *hostaddress portnumber*

This strings indicate the IP address (or DNS name) of a raw TCP host and the UDP port that the TCP session to that host uses. As explained above the port number is optional.

User-definable TCP connection retry timeout

This feature enables you to set the maximum length of time a MAX tries to complete a connection with a host IP address before proceeding to the next address on the list provided by the DNS server. Previously, this timeout length was always 170 seconds, which is longer than some client software will permit before the client software times out.

When a terminal server attempts to connect to a host through a MAX, and a DNS server is used, the DNS server may supply a list of host IP addresses in response to the query from the MAX. The DNS server simply provides the list and has no way of determining whether the hosts at the addresses are available or how long it may take for them to respond.

If the DNS List Attempt feature is enabled, the MAX will attempt to connect to the first address on the list until the timeout expires. Previously, the timeout default value was 170 seconds, which means that the MAX would attempt to connect to the first address on the DNS list for that length of time. Some client software for the terminal server user times out before 170 seconds elapse. When the client software times out, the connection is dropped by the client and none of the remaining addresses on the DNS list are tried. The connection may never be successful because each time it is retried and the DNS server provides the same list of addresses, the MAX starts at the top of the list again and attempts to make the same connection that was previously unsuccessful.

A new parameter, **`TCP Timeout`**, has been added to the Ethernet Configuration Profile, shown in Figure 5. Set this parameter to a value between 1 and 200 to specify the TCP retry time so that connections to additional host addresses can be attempted, if necessary, before the client software times out. When the timeout value expires and the connection to an address is unsuccessful, the MAX will proceed to try the next IP address on the list for the length of time specified in TCP timeout.

Choosing a value for TCP Timeout:

Setting the TCP timeout parameter depends on the characteristics of the TCP destination hosts. For example, if the destinations are on a local network under the same administrative control as the MAX and are lightly loaded, then a short timeout (a few seconds) may be reasonable because a host that does not respond within that interval is probably down.

A longer timeout is appropriate if the environment includes servers with

- longer network latency times
- high loads on the net or router
- characteristics of the remote hosts are not well known

Values of 30 to 60 seconds are common in UNIX TCP implementations.

The default value, zero, specifies that the MAX operate as previously and attempt for a maximum of 170 seconds to connect to each address on the list, until a connection is successful or the connection is dropped.

Figure 5. TCP timeout parameter in Ethernet Mod Config menu

```
90-A00 Mod Config
  RIP Summary=Yes
  ICMP Redirects=Accept
  BOOTP Relay...
  DNS...
  Multicast...
  Auth...
  Accounting...
  RADIUS server...
  Log...
  PPTP Options...
  Modem Ringback=Yes
  SNTP Server...
  Stack Options...
  UDP Cksum=No
  TCP Timeout=0
  Adv Dialout Routes=Always
```

TCP timeout parameter reference

Description: This parameter specifies the length of time during which a MAX will attempt to connect to an IP host in the list provided by the DNS server.

Since the first host on the list may not be available, the timeout should be short enough to allow the MAX to go on to the next address on the list before the client software times out.

This feature applies to all TCP connections initiated from the MAX, including telnet, rlogin, tcp-clear, and the TCP portion of DNS queries.

Usage: To set the timeout value, select TCP timeout and type the number of seconds the MAX can attempt a connection to an IP address on the DNS list.

The range of values for TCP timeout 0 to 200 seconds. This specifies the number of seconds after which the MAX will stop attempting to connect to an IP address and will proceed to the next address on the list. (but as noted below, other limits already in the system may terminate TCP retries after about 170 seconds). The number of start-connection messages the MAX will send is fixed, however.

Note: When the MAX has sent the maximum number of messages to an address on the DNS list it will stop attempting to make a connection to that address, even if the maximum time set in DNS Timeout has not yet elapsed.

The default for DNS Timeout is 0. If **`TCP timeout=0`**, the MAX will retry the connection to the address at increasingly large intervals until it sends the maximum number of start-connection messages. This takes approximately 170 seconds, but can take longer if the MAX is running large number of other tasks. If the client software times out before the MAX makes a connection or proceeds to the next address on the DNS list, the physical connection is dropped.

The List Attempt parameter in the DNS submenu of the Mod Config menu in the Ethernet Profile must be enabled. This permits the MAX to attempt the IP addresses. On a list, if the DNS server provides such a list. The List Attempt parameter does not apply if Telnet and Immediate Telnet are both disabled.

`Ethernet Profile: Ethernet/Mod Config`

MAX supports new CSLIP Auto Detect parameter

Previously, when you brought up a SLIP session, compression was the default. Now, you can bring up a SLIP session and choose no compression until you receive a VJ Compressed CSLIP packet. At that point, the MAX switches automatically to VJ compression mode.

CSLIP Auto Detect

Description: Enables and disables auto-detect of VJ Compressed CSLIP packet. Previously, when you brought up a SLIP session, compression was the default. Now, you can bring up a SLIP session and choose no compression until you receive a VJ Compressed CSLIP packet. At that point, the MAX switches automatically to VJ compression mode.

Usage: The CSLIP Auto Detect parameter has two options:

- Yes: VJ Compression is always on for all CSLIP packets.
- No: Compression is off for CSLIP packets until the MAX receives a VJ Compression CSLIP packet. When this occurs, the MAX starts VJ compression of all subsequent CSLIP Packets. This is the default.

Note: Depends on VJ Comp parameter. Applies only if VJ Comp=Yes, otherwise CSLIP Auto Detect=NA.

Example: CSLIP Auto Detect=Yes

Location: Mod Config > Tserv options

More information provided for SLIP connections

You can now specify the kind of information the MAX reports when a user connects over a SLIP link.

You can now specify the information that the MAX reports to users when they establish a Serial Line Internet Protocol (SLIP) connection.

Previously, the MAX always reported the following information whenever a user connected:

```
Entering SLIP Mode
IP address is 192.1.1.1
MTU is 1500
```

Below is an example of the kind of information the MAX can now report:

```
Entering SLIP Mode
IP address is 192.1.1.1
MTU is 1500
Netmask: 255.255.255.0
Gateway: 192.168.6.181
```

The Netmask label identifies the subnet mask the MAX is using. The Gateway label identifies the MAX unit's IP address. The sections that follow describe these parameters.

New parameters

The MAX interface includes three new parameters to support this feature:

- SLIP Info
- IP Netmask Msg
- IP Gateway Adrs Msg

These parameters are described below.

SLIP Info

Description: Specifies the type of information the MAX reports to SLIP users.

Usage: Specify one of the following values:

- Basic (the default)
  Specifies that the MAX only reports the SLIP user's IP address and the Maximum Transmission Unit (MTU).
- Advanced
  Specifies that the MAX reports the SLIP user's IP address, the MTU, the Netmask, and the Gateway to SLIP users. Note that the gateway is the MAX unit's IP address.

Location: Ethernet>Mod Config>TServ Options

IP Netmask Msg

Description: Specifies the text the MAX displays before the netmask field in the SLIP session startup message.

Usage: Specify a a text message. You can enter up to 64 characters. The default is `Netmask:`.

Dependencies: Keep this additional information in mind.

- IP Netmask Msg does not apply unless you set SLIP Info to Advanced.

Location: Ethernet>Mod Config>TServ Options

IP Gateway Addr Msg

Description: Specifies the text the MAX displays before the MAX IP address field in the SLIP session startup message.

Usage: Specify a a text message. You can enter up to 64 characters. The default is `Gateway:`.

Dependencies: Keep this additional information in mind.

- IP Gateway Addr Msg does not apply unless you set SLIP Info to Advanced.

Location: Ethernet>Mod Config>TServ Options

Multilink or MP+ call now can span multiple MAX units

Multiple MAX units can now be configured to form a stack, or group of MAX units, that allows a Multilink PPP (MP) or MP+ call to span the MAX units in the stack. This feature, previously added to the MAX 4000, is now available for other MAX units.

Call spanning using a stack configuration can be effective when:

- A MAX running MP+ is asked for another phone number, and has no available lines
- A rotary hunt-group uses the same phone number to access multiple MAX units, making it impossible to assume that a subsequent call is answered by the same MAX as the original call

MP/MP+ call spanning is protocol independent, and should work with all protocols supported by the MAX.

Note: Stacking requires any MP caller to use the MP endpoint discriminator. The same is true of MP+. All Ascend products and most other products that support MP or MP+ use an endpoint discriminator, but the specification for MP does not require it.

How MP/MP+ call spanning works

A stack is a group of MAX units that have the same stack information, and are on the same physical LAN. There is no *master* MAX; the MAX units in the stack use an Ethernet multicast packet to locate each other.

Multicast packets usually cannot cross a router, so the MAX units in a single stack must be on the same physical LAN. MAX units running in a stack can generate fairly high levels of network traffic, which is another reason to keep them on the same physical LAN.

Bundle ownership

Although MAX stacks do not have a master MAX, each MP/MP+ bundle has a bundle owner. The MAX that answers the first call in the MP/MP+ bundle is the *bundle owner.* If a bundle spans more than one MAX in a stack, an exchange of information flows between the MAX units in the bundle.

Stacking requires an endpoint discriminator. Every MP/MP+ call that comes to any member of the stack is compared to all existing MP/MP+ calls in the MAX stack to determine whether it is a member of an existing bundle. If the call belongs to an existing bundle, the MAX that answered and the bundle owner exchange information about the bundle. Furthermore, the MAX that answered the call forwards all incoming data packets over the Ethernet to the bundle owner.

Outgoing data

To balance the load among all available WAN channels, outgoing data packets for the WAN are assigned to available channels in a bundle on a rotating basis. If an outgoing packet is assigned to a channel that is not local to the bundle owner, the bundle owner forwards the packet over the Ethernet to the MAX that owns the non-local channel.

Real and stacked channels

For the purpose of this description, *real* channels are those channels that connect directly to the MAX that owns the bundle. *Stacked* channels connect to a MAX that transfers the data to or from the MAX that owns the bundle.

For example, assume the initial call of an MP/MP+ bundle connects to MAX #1. This connection is a *real* channel. Next, the second call of the bundle connects to MAX #2. This connection is a *stacked* channel. MAX #1 is the bundle owner, and it manages the traffic for both channels of the bundle. MAX #2 forwards any traffic from the WAN to MAX #1, for distribution to the destination. See Figure 6.

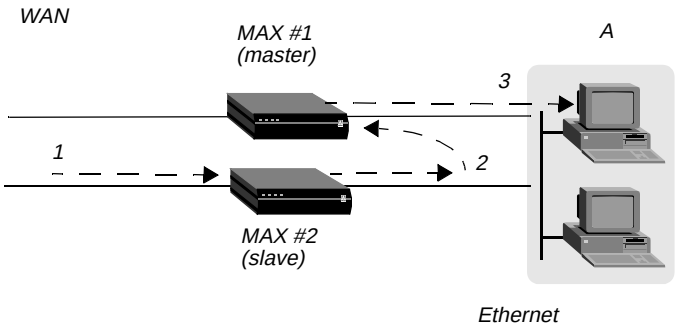


Figure 6. Packet flow from the slave channel to the Ethernet

Note: This graphic does not illustrate traffic from the master MAX. WAN traffic received on the master channel by MAX#1 is forwarded directly to the destination.

Likewise, MAX#1 receives all Ethernet traffic destined for the bundle, and disperses the packets between itself and MAX#2. See Figure 7. MAX#1 forwards some of the packets across the WAN through a real channel. MAX#2 sends the rest of them through a stacked channel.

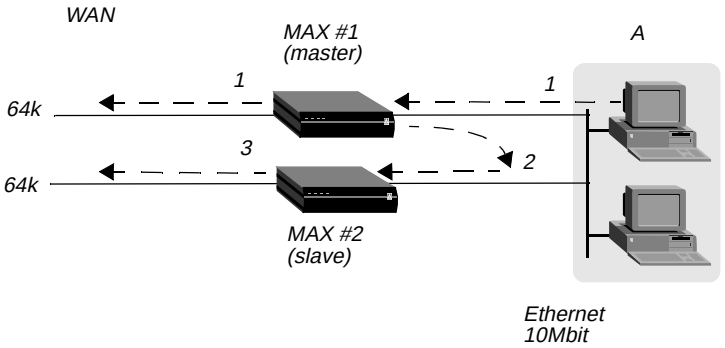


Figure 7. Packet flow from the Ethernet

Connection profiles not shared within a stack

A stack does not support sharing of local Connection profiles between the MAX units in the stack. Every MAX in the stack that is set up to use internal authentication must retain all authentication information for every call. You can eliminate this requirement by using a centralized authentication server, such as RADIUS.

Phone numbers for new MP+ and MP-with-BACP channels

When a MAX has to add a channel for a MP+ or MP-with-BACP call, it provides a local phone number for the new channel. However, sometimes the MAX that answers the call cannot provide a local phone number for the additional channel, because all the channels that connect directly to it are busy. In that case, the MAX requests other members of the stack to supply a phone number for the additional channel.

An MP call does not pass phone numbers when it adds a channel. The originator of the call must know all of the possible phone numbers to begin with.

If each MAX in the stack is accessed through a different phone number, the originator of the call must know all of the possible phone numbers. An alternative in this instance is to use BACP or MP+ to obtain the phone number of a MAX with a free channel.

Performance considerations for MAX stacking

There is no limit to the number of *stacked* channels in single call or in a stack of MAX units, other than the limit for each individual MAX. The MAX 4000, MAX 2000, and MAX 1800 each support up to 40 stacked channels. The MAX 200 Plus supports up to three stacked channels. A MAX can handle *n* real channels and *n*/3 *stacked* channels.

There is no theoretical limit to the number of MAX units in a stack, other than performance considerations. Since all data from stacked channels crosses the LAN, performance could suffer with a large number of MAX units in the stack and many stacked channels in use.

Performance overhead increases when stacked bundles span multiple boxes. In a bundle of 6 channels, 4 of which are real and 2 are stacked, the overhead is the actual bandwidth of the two stacked channels (2 x 64 = 128K). The actual payload data of the 6 channels with a 2:1 data compression is 6 x 2 x 64 = 768K. The overhead is 128 over 768, or 16%. In a two-channel bundle with one real and one stacked channel, with the same compression, the overhead is 25%.

Take into account that you do not know ahead of time how many bundles will span the stack, or how many multi- or single-channel calls you are going to get. You can base an estimate on your traffic expectations. But in most situations, the majority of bundles will be on a single MAX, for which there is no overhead.

Suggested LAN configurations

Calculations like the ones mentioned above show that when your MAX stack handles 82 single-channel calls, 41 two-channel stacked calls, and 41 two-channel nonstacked calls, the Total Ethernet usage is approximately 5116Kbps. Since Ethernet capacity generally does not achieve more than 50% utilization, this configuration uses up the available Ethernet bandwidth.

The total number of channels in this configuration is 246. Therefore, a stack of three MAX units, each having three T1 lines with this usage profile, utilizes all of the Ethernet bandwidth.

The basic limitation from the above examples is the speed of the LAN. One way to increase the speed of your LAN is to attach each MAX to a separate port of a 10/100 Ethernet switch, then use a 100Mbps connection to the backbone LAN. This allows each MAX to utilize up to a full 10Mb Ethernet and the entire stack combined can generate up to full 100Mb of Ethernet data.

Once again assuming that the 100Mpbs is saturated at 50% usage, we can now use up to 51200Kbps of bandwidth, or 10 times more than in the example above. Note that the success of this strategy depends on limiting stacked channels per MAX to the n/3 limit mentioned above.

Suggested hunt-group configurations

Whenever you have MAX units in a stack, it is important to limit the number of multichannel calls that are split between the MAX units. The following suggested configurations reduce the overhead for a multichannel call by keeping as many channels as possible on the same MAX.

MP+ and MP-with-BACP calls

Figure 8 shows the suggested hunt-group setup for a typical MAX stack that receives only PPP, MP+, or MP-with-BACP calls. Each MAX has three T1 lines. All the T1 lines in a MAX share a common phone number and they are in a hunt-group that does not span MAX units. The illustration shows these three local hunt-groups with phone numbers 555-1212, 555-1213, 555-1214. In addition, a global hunt-group, 555-1215 spans all the T1s of all the MAX units in the stack.

Users that access the MAX dial 555-1215, the global hunt-group number. The telephone company has set up the global hunt-group to distribute incoming calls equally among the MAX units. Namely, the first call dialing 555-1215 goes to MAX#1, the second call to MAX #2, and so on. If you use this configuration, you must configure each of the MAX unit's Line profiles with the local hunt-group numbers. For example, for MAX #1 in Figure 8, you would set the Ch *n* # parameters to 12 (the last two digits of the 555-1212 hunt-group number).

You can achieve the same distribution without a global hunt-group by having one third of the users dial 555-1212, one third dial 555-1213, and one third dial 555-1214. You can leave the Ch *n* # parameters at their default setting (null) if you do not have a global hunt-group.

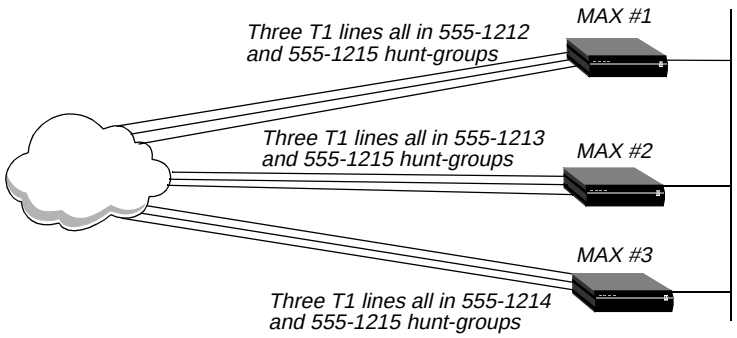


Figure 8. Hunt-groups for a MAX stack handling both MP and MP+ calls

Viewing Figure 8, suppose an MP+ call is connected to MAX #1. When that call needs to add a channel, it requests an add-on number from the MAX, and the MAX returns *12* (for 555-1212) as long as a channel in the local T1 lines is available. This means the bundle will not span multiple MAX units as long as a channel is available in the local hunt-group.

The Figure 8 configuration tends to break down if MAX units receive MP-without-BACP calls. Spreading the calls across the MAX stack (by dialing the global hunt-group) results in

the worst possible performance, because MP-without-BACP must know all of the phone numbers before the caller places the first call.

MP-without-BACP calls

Figure 9 shows a site that supports only MP-without-BACP calls. For this site, the telephone company has set up a global hunt-group that first completely fills MAX #1, then continues to MAX #2, and so on. This arrangement tends to keep the channels of a call from being split across multiple MAX units, keeping overhead low.

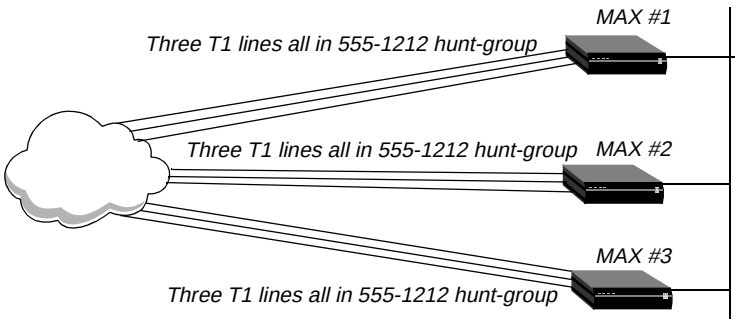


Figure 9. Hunt-groups for a MAX stack handling only MP-without-BACP calls

MP+ calls and MP calls with or without BACP

For a MAX that receives MP+ calls and MP calls with or without BACP, you can use a configuration similar to the one shown in Figure 8. In this case, however, you set up the global hunt-group differently than explained in "MP+ and MP-with-BACP calls." You set up the global hunt-group to help prevent MP-without-BACP calls from being split across multiple MAX units in the stack. As in "MP-without-BACP calls," calls dialing 555-1215 first completely fill the channels of MAX #1, then continues to MAX #2, and so on.

Both MP+ and MP callers dial the global hunt-group number to connect to the stack. Channels added to the MP+ and MP bundles are handled as explained in "MP-without-BACP calls," and "MP+ calls and MP calls with or without BACP." Be sure to set the Ch n # parameters as explained in "MP+ calls and MP calls with or without BACP."

MP+ and MP-with-BACP callers do not have to dial the global hunt-group numbers to connect. Only the MP-without-BACP callers need to dial the global hunt-group. You can achieve an even distribution of MP+ and MP-with-BACP calls by having one third dial 555-1212, one third dial 555-1213, and one third dial 555-1214. You can leave the Ch *n* # parameters at their default setting (null) in this situation.

Configuring a MAX stack

To configure a MAX stack, proceed as follows for each MAX in the stack:

1 Open the Ethernet > Mod Config menu, and select Stack Options, as shown in the following sample menu:

```
90-A** Mod Config
 RADIUS Server
```

```
Log
ATMP
Modem Ringback=Yes
AppleTalk
SNTP Server
>Stack Options...
UDP Checksum=No
```

When you press Enter, the Ethernet > Mod Config > Stack Options menu appears. For example:

```
90-A** Mod Config
>Stack Options...
Stack Enabled=Yes
Stack Name=astack
UDP Port=5151
```

2 Set Stack Enabled to Yes (Stack Enabled=Yes).

3 Set the Stack Name parameter to a unique name for the stack.

A stack name is 16 characters or less. This is the name members of a stack use to identify other members of the same stack. The stack name must be unique among all MAX units that communicate with each other, even if they are not on the same LAN.

If a MAX receives calls from two MAX units on different LANs, and the two units are members of different stacks with the same stack name, the MAX receiving the calls assumes the two MAX units with the same stack name are in the same bundle.

Note: Multiple stacks can exist on the same physical Ethernet LAN if the stacks have different names.

4 Specify the UDP port.

This is a reserved UDP port for intrastack communications. The UDP port must be identical for all members of a stack, but is not required to be unique among all stacks.

Disabling a MAX stack

To disable a stack, specify Stack Enabled=No for each of the MAX units in the stack.

Adding and removing a MAX

You can add a MAX to an existing stack at any time without rebooting the MAX or affecting stack operation. Since a stack is a collection of peers, none keeps a list of the stack membership. The MAX units in a stack communicate when they need a service from the stack.

Removing a MAX from a stack requires care, because any calls using a channel between the MAX to be removed and another MAX in the stack could be dropped. There is no need to reboot a MAX removed from a stack.

Parameter Reference

This release adds two new parameters: Stack Enabled and Stack Name.

Stack Enabled

Description: Stack Enabled enables MP and MP+ call spanning for the MAX. When stack Enabled=Yes, a *stack*, or group of MAX units that have the same stack information and are on the same physical LAN. The MAX units in the stack use an Ethernet multicast packet to locate each other. Once a stack is created, every MP/MP+ call that comes to any member of the stack is compared with MP/MP+ calls to other members of the stack to determine if it is part of an already existing bundle.

If you disable this parameter, all channels in an MP or MP+ bundle must exist on a single MAX. If you enable this parameter, MP and MP+ bundles can span any of the MAX units in the stack. Note that MP and MP+ bundles that span MAX units cause additional traffic on the Ethernet as the MAXs in the bundle route packets between them.

Stacking requires an endpoint discriminator. Every MP/MP+ call that comes to any member of the stack is compared to all existing MP/MP+ calls in the MAX stack to determine whether it is a member of an existing bundle. If the call belongs to an existing bundle, the MAX that answered and the bundle owner exchange information about the bundle. Furthermore, the MAX that answered the call forwards all incoming data packets over the Ethernet to the bundle owner.

Usage: Select Yes or No.

- Yes enables MP and MP+ call spanning.
- No disables MP and MP+ call spanning.

Location: Ethernet > Mod Config > Stack Options

See Also: Stack Name

Stack Name

Description: Stack Name defines a unique name for a MAX stack. This is the name members of a stack use to identify other members of the same stack. The stack name must be unique among all MAX units that communicate with each other, even if they are not on the same LAN.

If a MAX receives calls from two MAX units on different LANs, and the two units are members of different stacks with the same stack name, the MAX receiving the calls assumes the two MAX units with the same stack name are in the same bundle.

Multiple stacks can exist on the same physical Ethernet LAN if the stacks have different names.

Usage: Press Enter to open a text field, then type a unique name for the MAX stack, up to a maximum of 16 characters.

Location: Ethernet > Mod Config > Stack Options

See Also: Stack Enabled

Updated Microsoft Callback Control Protocol support

This release adds support for Microsoft's CallBack Control Protocol (CBCP). CBCP is a Link Control Protocol (LCP) option negotiated at the beginning of Point to Point Protocol (PPP) sessions. CBCP authenticates callers by means of user names and passwords, and offers additional security to enable the MAX to ensure that connections are to known users.

Introduction

Microsoft developed CBCP to address a need for greater security with PPP connections. The standardized callback option defined in RFC 1570 has a potential security risk because the authentication is performed after the callback. CBCP callback like Ascend's proprietary callback, occurs after authentication, leaving no potential security hole.

CBCP also offers features not available with the standard callback defined in RFC 1570. The client side supports a configurable time delay to allow users to initialize modems or enable supportive software before the MAX calls the client. You can configure the MAX with a phone number to use for the callback, or you can configure it to allow the client to specify the phone number used for the callback.

Currently, Microsoft's Windows NT 4.0 and Windows 95 software support client-side authentication using CBCP. The MAX now supports a CBCP central-site solution.

Ascend's implementation of CBCP

CBCP is an option negotiated during the LCP negotiation of a PPP session. While support for CBCP is configured systemwide on the MAX, not every connection must negotiate its use. Parameters have been added to the Answer Profile under Ethernet > Answer > PPP Options, and to each Connection Profile under Ethernet > Connections > Encaps Options. The calling and called sides of a PPP session initiate authentication after acknowledging that CBCP is to be used.

Note: Currently, the MAX does not initiate LCP negotiation of CBCP. The MAX responds to *caller* requests to configure CBCP.

The MAX employs the user name and password to link a caller with a specific Connection profile or RADIUS User profile. Configured CBCP parameters in that Connection profile specify variables for the callback. If, at any point, the client and the MAX disagree about any CBCP variables, the MAX might drop the connection.

Both sides of the connection must agree on whether the callback phone number is supplied by the client or by the MAX. A new trunk group parameter, configured on the MAX, supplies a trunk group that is prepended to phone numbers when supplied by the client.

Negotiation of CBCP

Following are the steps from initial connection to MAX callback:

1 Caller connects to MAX.

2 LCP negotiations begin.
Caller and MAX must agree to use CBCP. Otherwise, the MAX terminates the connection.

3 After successful LCP negotiation, both sides have acknowledged that CBCP will be used, and CBCP begins after authentication.

4 Caller authenticates itself to MAX. If authentication fails, the MAX terminates the connection.

5 The MAX verifies that the profile has CBCP Mode set. CBCP begins.

6 The MAX sends a request to determine if a callback is to occur. The caller's configuration must match the CBCP Mode value on the MAX.

The client also supplies to the MAX the number of seconds it should delay before initiating the callback, and, if applicable, the phone number.

7 If both sides agree on which phone number the MAX will dial, the client clears the connection.

8 The MAX delays the callback on the basis of the previous negotiation.

9 The MAX dials the client, by applying information from the same profile used in previous negotiation.

Configuring Microsoft's CBCP to use a Connection Profile

To configure CBCP to work with a Connection profile:

1 Open the Ethernet > Answer > PPP Options menu.

2 Set CBCP Enable = Yes.

3 Open the Ethernet > Connections > *Any Connection profile* > Encaps Options menu.

4 Set CBCP Mode to the callback mode to be offered the caller.

5 If the caller is supplying the phone number, set CBCP Trunk Group to the value (4-9) that the MAX prepends to the number when calling back.

6 Save your changes.

New parameters

The following parameters have been added to the VT100 interface:

CBCP Enable

Description: Specifies how the MAX responds to caller requests to support CBCP.

Usage: Press Enter to cycle through the choices.

- Yes specifies the MAX will positively acknowledge, during LCP negotiations, support for CBCP.
- No specifies the MAX will reject any request to support CBCP.

  No is the default.

Location: Ethernet > Answer > PPP Options

See Also: CBCP Mode, CBCP Trunk Group

CBCP Mode

Description: Specifies what method of callback the MAX offers the incoming caller.

Usage: Press Enter to cycle through the choices. You can specify one of the following settings:

Setting	Description
No Cback	Applies for Windows NT or Windows 95 clients who must not be called back. Because CBCP has been negotiated initially, the Windows clients must have validation from the MAX that no callback is used for this connection.
User Num	Specifies that the caller will supply the number the MAX uses for the callback.
Prof Num	Specifies the MAX will use the number in Ethernet > Connections > *Any* Connection profile > Dial # for the callback
User Num or No Cback	Specifies that the caller has the option of either supplying the number to dial or specifying that no callback is used for the call. If no callback is chosen, the call will not be disconnected by the MAX.

Dependencies: CBCP Mode applies only if CBCP is successfully negotiated for a connection. Encaps=PPP or MPP or MP.

Location: Ethernet > Connections > *Any* Connection Profile > Encaps Options

See Also: CBCP Enable, CBCP Trunk Group

CBCP Trunk Group

Description: Assigns the callback to a MAX trunk group. This parameter is used only when the caller is specifying the phone number the MAX uses for the callback. The value in CBCP Trunk Group is prepended to the caller-supplied number when the MAX calls back.

Usage: Press Enter to open a text field. Then type a number from 4 to 9. The default is 9.

Dependencies: CPCP Trunk Group applies only if CBCP is negotiated for a connection. Encaps=PPP or MPP or MP.

Location: Ethernet > Connections > *Any* Connection Profile > Encaps Options

See Also: CBCP Enable, CBCP Mode

Configuring Microsoft's CBCP to use a RADIUS Profile

New RADIUS attributes support CBCP in User profiles. Ascend-CBCP-Enable specifies how the MAX responds to caller requests to support CBCP. AScend-CBCP-Mode specifies the method of callback the MAX offers the incoming caller. Ascend-CBCP-Trunk-Group assigns the callback to a MAX trunk group.

Note: Make sure you set CBCP Enable=Yes in the Ethernet > Answer > PPP Options menu.

Ascend-CBCP-Enable (112)

Description: Specifies how the MAX responds to requests by callers to support CBCP.

Usage: Specify one of the following settings:

- CBCP-Enabled (0)—Specifies that the MAX will positively acknowledge, during LCP negotiations, support for CBCP.
- CBCP-Not-Enabled (1)—Specifies that the MAX will reject any request to support CBCP.

See Also: Ascend-CBCP-Mode, Ascend-CBCP-Trunk-Group

Ascend-CBCP-Mode (113)

Description: Specifies what method of callback the MAX offers the incoming caller.

Usage: Specify one of the following values:

- CBCP-No-Callback (1)—Applies for Windows NT or Windows 95 clients who must not be called back. Because CBCP has been negotiated initially, the Windows clients must have validation from the MAX that no callback is used for this connection.
- CBCP-User-Callback (2)—Specifies that the caller will supply the number the MAX uses for the callback.
- CBCP-Profile-Callback (3)—Specifies that the MAX will use the number in Ascend-Dial-Number for the callback
- CBCP-User-Or-No (7)—Specifies that the caller has the option of either supplying the number to dial or specifying that no callback is used for the call. If no callback is chosen, the call will not be disconnected by the MAX.

Dependencies: Ascend-CBCP-Mode applies only if CBCP is successfully negotiated for a connection.

See Also: Ascend-CBCP-Enable, Ascend-CBCP-Trunk-Group

Ascend-CBCP-Trunk-Group (115)

Description: Assigns the callback to a MAX trunk group. This attribute is used only when the caller is specifying the phone number the MAX uses for the callback. The value in Ascend-CBCP-Trunk-Group is prepended to the caller-supplied number when the MAX calls back.

Usage: You can specify a number between 4 and 9, inclusive. The default is 9.

Dependencies: Ascend-CBCP-Trunk-Group applies only if CBCP is negotiated for a connection.

See Also: Ascend-CBCP-Enable, Ascend-CBCP-Mode

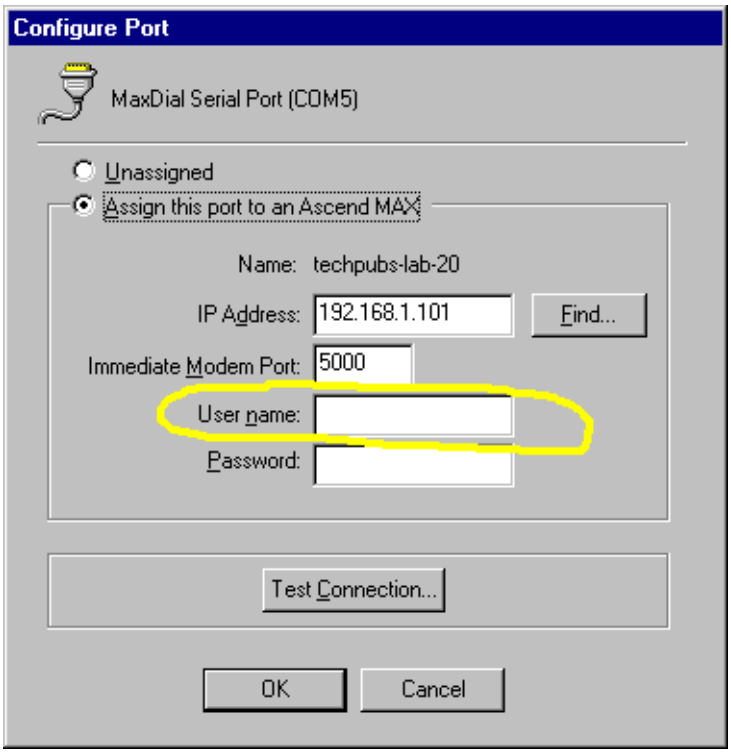
MAXDial/IP support for immediate modem call restriction

MAXDial now supports the call restriction modes for immediate modem service. The immediate modem service in the MAX terminal server now supports call restriction modes.

MAXDial has been updated to connect to a MAX that has the updated immediate modem service.

In the MAX system software, immediate modem service now supports three authentication modes: "none", "global password", or "user". When the immediate modem service is configured in "user" mode, the MAXDial software must be configured to supply a user name as well as a password to use the digital modems for dialing out. To implement this change, the MAXDial Ports Control Panel has a new User Name field.

If the immediate modem service is configured in the "none" or "global password" mode, the User Name field may be empty. If the immediate modem service is configured in "user" mode, MAXDial displays a "login name required" error message if its User Name field is empty and the user attempts to dial out on a digital modem.



Support added for multiple host selection

You can now specify up to three authentication hosts for Defender authentication, so that these hosts can serve as backups for each other. Previously, you could only specify one authentication host, although the Defender authentication mechanism itself allowed multiple hosts.

How it works:

There are three major stages in authentication using AssureNet Pathways' Defender. The MAX' behavior will depend upon the stage the call dialing the MAX was in when the connection with the host is lost.

Table 10. Token card authentication

Stage	Description	MAX Behavior at this stage
1	Usually a short time after the caller has connected to the MAX and before the MAX has received the first prompt from the authentication host. The Defender server provides the text of the prompts or challenges, and the MAX passes them through to the caller.	Calls in Stage 1 are preserved if an authentication host is unavailable or loses its connection. This might be the case when the very first caller is authenticating with Defender after the router boots up, and the first authentication host is unavailable. The Defender authentication code in the router will try the second and third hosts in order to authenticate the user.
2	During the time the caller is interacting with the authentication host, but before the authentication sequence is complete. The Defender uses a challenge-response protocol, with a token card to provide the responses.	Calls in Stage 2 are never preserved if an authentication hosts loses its connection. Defender has no mechanism for having one authentication server take over for another if the first loses connection in the middle of a state.
3	When the caller has completed authentication and is interacting with the MAX normally (either asynchronously or framed).	Callers in Stage 3 are not dropped by the router since their calls are already authenticate. However, because the host on which they authenticated is no longer available, their logout time will not be sent (as would be the case if the host had remained connected). Defender provides no mechanism to notify one authentication host when a user call that was authenticated by another host is terminated.

When no authentication host is available

When a MAX can not establish contact with any of the authentication hosts in the list, all sessions are dropped, including calls in Stage 1.

If a caller who has been disconnected tries again to make a connection, the MAX will begin again the process of connecting to authentication hosts on the list until it either succeeds or has tried every host in the list.

User interface changes

You can specify up to three authentication hosts in the Auth submenu of the Ethernet Configuration Profile using the Auth Host #1, #2, and #3 parameters. Previously these were N/A when Auth-DEFENDER. You can one of the Auth Host parameters, as shown in the example for Auth Host #2. The authentication process checks for null IP address and does not attempt to use a null IP address.

```
90-100 Mod Config
 Auth
 >Auth=Defender
  Auth Host #1=137.175.80.62
  Auth Host #2=0.0.0.0
  Auth Host #3=137.175.80.24
  Auth Port=2626
  Auth Src Port=0
  Auth Timeout=30
  Auth Key=
```

Changes to syslog messages

This EOI introduces a set of syslog messages reporting the status of the Defender authentication subsystem. The new syslog messages are reported with "LOG_DEFAULT" and "LEVEL_INFO" priorities. The following lines exemplify the new syslog messages.

```
Nov 14 15:59:34 137.175.85.20 ASCEND: AuthHost 137.175.81.24
Activated

Nov 14 15:51:10 137.175.85.20 ASCEND: AuthHost 137.175.81.24
Fails auth

Nov 14 15:51:10 137.175.85.20 ASCEND: AuthHost 137.175.80.24
Refuses connect

Nov 14 16:03:05 137.175.85.20 ASCEND: AuthHost 137.175.81.24
Closed connection

Nov 14 16:05:59 137.175.85.20 ASCEND: AuthHost 137.175.81.24
Address Changed

Nov 14 16:06:31 137.175.85.20 ASCEND: AuthHost 137.175.81.24
New Authmethod
```

Message format

All Defender syslog messages report the standard Ascend header plus Defender-formatted detail: "AuthHost xx.yy.zz.aa Statusx" where "Statusx" has the values shown in the following table*:*

Table 11. Status indicators

Status	Description
Activated	A Defender Host has been found and a connection successfully established. This state is reported when an authentication session is active and ready to authenticate.
Fails auth	A Defender Host has been found, but the router and the Defender authentication server do not agree on their mutual authentication key.
Refuses connect	The host either is not responding at all or has no active Defender Server running.
Closed connection	An active Defender authentication server has ended its connection. This would reflect either a failure of the server software or explicit request by an administrator for the server to shutdown.
Address Changed	The MAX administrator has changed the IP address, the port number, or the authentication key of the active authentication server. This forces the Defender authentication subsystem to close its connection with the active server and start searching for a new one.
New Authmethod	The MAX administrator has changed the authentication method from "DEFENDER" to something else, causing the Defender authentication subsystem to break an active connection.

Limiting terminal server access per user

This feature allows the MAX to limit particular users to a subset of terminal server commands.

The Framed Only parameter in the Answer profile and the Connection profiles allows administrators to limit particular users to the PPP, SLIP, CSLIP, and Quit commands in the MAX terminal server interface.

Configuring per-user access to terminal server commands

You can configure per-user access to the terminal server commands in the Answer profile or in the Connection profile:

- The Answer profile affects users who do not have a Connection profile, users with a Name/Password profile, or RADIUS-authenticated users whose connections are built in part with the Answer profile
- The Connection profile only affects individual users connecting to the MAX using a particular Connection profile

To configure per-user access to the terminal server:

1 Select Ethernet > Answer > Session Options *or*
Ethernet > Connections > *a Connection profile* > Session Options

2 Specify one of the following values for Framed Only:

– No (the default)

Specifies that terminal server users connecting through this profile have unlimited access to the terminal server commands.

– Yes

Specifies that terminal server users connecting through this profile only have access to the PPP, SLIP, CSLIP, and Quit terminal server commands.

3 Save and exit the profile.

If a user restricted to these commands tries to execute any other terminal server command, the MAX displays the following message:

Unauthorized Terminal Server Command.

Parameter reference

This section describes the new MAX parameter.

Framed Only

Description: Specifies whether the user is allowed access to all the terminal server commands or to a subset of them.

Usage: Specify one of the following values:

- No (the default)

  Specifies that terminal server users connecting through this profile have unlimited access to the terminal server commands.
- Yes

  Specifies that terminal server users connecting through this profile only have access to the PPP, SLIP, CSLIP, and Quit terminal server commands.

Dependencies: Keep this additional information in mind:

- Framed Only has no affect if TS Enabled is set to No in the Ethernet > Mod Config > TServ Options submenu.

- PPP, SLIP, and CSLIP must be enabled in the Ethernet > Mod Config > TServ Options submenu before users can start a PPP, SLIP, or CSLIP session.

Location: Ethernet > Answer > Session Options
Ethernet > Connections > *any Connection profile* > Session Options

User-configurable call blocking after failed connection attempt

You can now block additional retry attempts after a specified number of failed connection attempts have been made, and control the length of time call blocking is in effect. Previously you could not automatically stop the Ascend unit from attempting to place an outgoing call on a connection that repeatedly fails.

Overview

When an Ascend unit attempts to make a connection and the attempt fails, the Ascend unit continues to attempt to complete the connection. The number of retry attempts allowed without using this new call blocking feature is very large; successive retries can cause excessive charges, congestion, and performance problems. This feature enables you to specify the number of unsuccessful attempts to place a call that an Ascend unit can make before blocking further attempts to make that connection. After the specified number of attempts have been made and failed, the blocking timer starts. The Ascend unit continues to block further calls for a the period of time you specify.

Configuring call blocking

1. Open the Session options submenu of the Connection Profile.
2. Select Block calls after and enter the number of retry attempts to allow the Ascend unit to make when placing a call.
3. Select Blocked duration and specify the length of time during which the Ascend unit will continue to block calls to number in the Connection Profile.

Note: This feature applies only to outgoing calls that are not answered by the far end. It does not apply to:

- incoming calls
- outgoing calls that connect and are immediately disconnected

Parameter reference

Two new parameters have been added to the Session submenu of the Connection Profile.

Block calls after

Description: Specifies how many unsuccessful attempts the Ascend unit will make before beginning to block outgoing calls.

Usage: Enter the number of connection attempts permitted before the Ascend unit blocks calls for the connection. The maximum number you can enter is 65535 (65535 attempts). The default is 0.

Location: Session Options submenu of the Connection Profile.

See Also: Blocked duration

Blocked duration

Description: Specifies the length of time in seconds during which the Ascend unit will block outgoing calls.

Usage: Enter the number of seconds for the Ascend unit to block all calls made to the connection. When this period has elapsed, the unit will again allows calls to this connection.

Location: Session Options submenu of the Connection Profile.

See Also: Block calls after

Specifying Shared Profiles per Connection Profile

Previously, you could configure shared profiles on a per-MAX basis by setting Ethernet > Mod Config > Shared Prof = Yes. This release adds a Shared Prof parameter to the Connection profile, enabling you to allow users to share specific profiles. You can also set a new RADIUS attribute to allow multiple incoming users to share a user file.

To use the new parameter or the new attribute, you must disable profile sharing on a per-MAX basis. That is, if you specify:

Ethernet > Connections > *Any Connection Profile* = Yes

or, in a RADIUS user profile:

Ascend-Shared-Profile-Enable = Shared-Profile-Yes

You must also specify:

Ethernet > Mod Config > Shared Prof = No

Shared Prof

Description: Enables multiple incoming callers to share a local Connection profile. Note that to apply shared profiles on a per-Connection-profile basis, you have to disable profile sharing on a system-wide basis by setting Ethernet > Mod Config > Shared Prof = No.

Usage: Press Enter to toggle between Yes and No.

- Yes specifies that multiple incoming calls can share a local Connection profile.

  The MAX must first authenticate the caller by applying the profile's Name and Recv PW parameters. If an incoming call has an IP address that conflicts with an existing caller IP address, or if the MAX would have to assign a conflicting IP address from the IP address pool, the MAX rejects the call.

- No specifies that multiple incoming calls cannot share a local Connection Profile.
  No is the default.

Dependencies: Shared Prof for Connection profiles applies only if you have disabled shared profiles for the MAX as a whole with Ethernet > Mod Config > Shared Prof = No

Location: Ethernet > Connections > *Any Connection Profile*

Ascend-Shared-Profile-Enable

Description: Enables or disables sharing of a RADIUS user file for multiple incoming users.

Note: To apply Shared Profiles on a per RADIUS user profile basis, you have to disable profile sharing on a system-wide basis by setting Ethernet > Mod Config > Shared Prof = No on the MAX

Usage: You can specify one of the following settings:

- Ascend-Shared-Profile-Enable = Shared-Profile-Yes specifies that multiple incoming calls can share this RADIUS user profile.
- Ascend-Shared-Profile-Enable = Shared-Profile-No specifies that multiple incoming calls cannot share a local Connection Profile.
  The default value is Shared-Profile-No

Dependencies: For the Ascend-Shared-Profile-Enable attribute to apply, you must disable shared profiles for the MAX as a whole with Ethernet > Mod Config > Shared Prof = No.

Terminal server users can be forced to use unique profiles

The MAX can now force terminal server users to connect using unique profiles.

New parameter

Shared Prof

Description: The MAX can force terminal server users to connect using unique profiles.The Shared Prof parameter in the Ethernet>Mod Config profile or in a Connection profile specifies:

- whether multiple users can share a single Connection profile or a single RADIUS user profile *or*
- whether a single user can have multiple sessions active

This parameter enables multiple incoming calls to share a local Connection profile or a RADIUS users file with Connection profile parameters. Sharing a profile cannot result in two IP addresses sharing the same interface, so this parameter is typically used to share profiles when the caller is assigned an IP address dynamically, which ensures that each caller is assigned a unique address.

Usage: Specify Yes or No. No is the default.

- Yes means the MAX will allow more than one caller to share the same profile, provided that no IP address conflicts will result.
- No means the MAX will not allow shared profiles.

Note: This feature extends support for the Shared Prof parameter to terminal server users. If Shared Prof is set to No and a user attempts to log in to the MAX terminal server with the same username and password as an already active session, the following message is displayed and the MAX disconnects the user: `***Account Already In Use`

Dependencies: This parameter does not apply to Combinet links or connections that have hard-coded IP addresses. For the Ascend-Shared-Profile-Enable attribute to apply, you must disable shared profiles for the MAX as a whole with Ethernet > Mod Config > Shared Prof = No.

Location: Ethernet > Mod Config, Ethernet > Connections > any profile

See Also: Encaps, Name, Pool # Count, Pool # Start, Recv PW

CR-LF characters act as a single CR in terminal server

The MAX terminal server treats Carriage Return-Line Feed characters as a single Carriage Return.

The MAX terminal server treats a CR-LF pair as a single CR instead of as 2 CRs. A CR-LF pair acts correctly as the end of a line in the terminal server. If a line ends with a CR, the MAX processes it as the end of the line; if it then sees a LF, the MAX ignores it. This change should be transparent to most users because the MAX still recognizes both a CR and a LF as the end of a line.

IP routing features

Changes to the IP router

Changes have been made to Ascend units' IP routing. These include changes to the routing table display and to diagnostic command output.

Changes have been made to the Ascend unit's IP routing stack which improve performance add additional support for multicast routing. These changes include:

- User interface changes
- Diagnostic mode changes
- Changes to Secure Access Firewall operation

In addition, Ascend units now conform more closely to RFC1812 (Requirements for routers) section 5.

User interface changes

The `iproute show` command output has been modified.

A route has been added from the 127 network to the blackhole interface:

```
127.0.0.0/8         -               bh0     CP      0   0       0       59593
```

Packets routed to the blackhole interface are discarded silently.

Routes pointing to local machines are now labeled local. These include the following routes:

```
127.0.0.1/32        -               local   CP      0   0       0       59593
224.0.0.1/32        -               local   CP      0   0       0       59593
224.0.0.2/32        -               local   CP      0   0       0       59593
w.x.y.z/32          -               local   CP      0   0       0       59593
```

with a single w.x.y.z route for each local IP address.

Note that the routes to 224.0.0.1 and 224.0.0.2 are new routes. They represent the multicast addresses for all systems on the local subnet and all routers on the local subnet, respectively, and are never forwarded.

A new route has been added to a virtual interface called mcast. All multicast addresses (except for special addresses such as 224.0.0.1/32 and 224.0.0.2/32) point to the mcast interface:

```
224.0.0.0/4         -               mcast   CP      0   0       0       59593
```

Diagnostic mode changes

The ippacket diagnostic output has been changed.

Modified diagnostic messages

The wording has been changed on these errors:

Table 12. Modified diagnostic messages

Previous message	New message
IP: no ip address for this port	IP: received packet on unconfigured interface
IP: options: calling icmp_send(): type = %d code = %d	IP: options: sending icmp to %s, type = %d code = %d
IP: passed pkt length is short	IP: received frame too small to hold any IP header
IP: short IP header	IP: received packet with header size < 20 bytes
IP: version check failed	IP: received unknown IP version %d
IP: bootp packet	IP: received BOOTP packet
IP: NAT packet	IP: received NAT packet
IP: checksum failed	IP: received bad checksum

Table 12. Modified diagnostic messages (continued)

Previous message	New message
IP: no memory	IP: no memory, dropping packet

New diagnostic messages

The following messages have been added:

- IP: received packet too small to hold its IP header
- IP: received truncated IP packet
- IP: received 0 ttl

Deleted diagnostic messages

The following messages have been deleted:

- IP: passed pkt length is short
- IP: (pkt <MIN_ETHER_LEN) length check failed
- IP: (pkt >MAX_ETHER_LEN) length check failed
- IP: (pkt <=MAX_ETHER_LEN) length check failed
- IP: (pkt <=MAX_ETHER_LEN) is padded
- IP: short length check failed
- IP: IF wants gateway %s, but no route
- IP: route to gateway %s isn't direct
- IP: (next hop to it is %s)
- IP: no route to %s.
- IP: not forwarding
- IP: bad incoming ttl of zero!!!
- IP: ttl expired
- Bad checksum pkt at 0x%p
- IP: parse: not bcast
- IP: parse: source & dest if different
- IP: NAT Session not active
- IP: reassembly error
- IP: not joined
- IP: unused Pool address.

Changes to Secure Access Firewall operation

A minor change has been made in the way that a Secure Access Firewall deals with directed broadcasts. A directed broadcast, received as a unicast, will not be delivered locally if the firewall on the outbound interface would block that packet. Thus, if the firewall on the outbound interface is set up to block a packet, no one will receive it, including the Ascend unit.

Previously the packet would be routed by the Ascend unit to the outbound interface, where it would be dropped by the firewall.

Additional information

Keep in mind the following changes to the IP router functionality:

- The MAX now correctly drops certain packets that it would have previously passed. This includes broadcast pings with a source address of 127.0.0.2.
- The MAX did not screen on the source address as defined in RFC1812 4.2.2.11. It does now, except in the case of a source address of 0.0.0.0.

Alphanumeric IP pool names

This feature allows you to assign alphanumeric names to IP address pools.

The Ethernet > Mod Config > WAN Options submenu now contains a Pool Name parameter. You can use this parameter to assign pool names to each of the MAX IP address pools. Certain types of authentication, such as TACACS+, require alphanumeric pool names. When the MAX authenticates a PPP call using TACACS+, it uses the Pool Name to determine which address pool it should use to assign the caller an address. If the Pool Name is not present, or the MAX doesn't find a match, it then attempts to match the Pool Number.

Configuring IP address pool names

To assign a name to an IP address pool:

1 Open the Ethernet > Mod Config > WAN Options submenu.

2 Set Pool *n* Name to the name of the address pool, where *n* is the number of the address pool.

3 Exit and save the Mod Config profile.

Parameter reference

This section describes the new MAX parameter.

Pool Name

Description: Specifies the name of an IP address pool.

Usage: Specify a name. You can enter up to 10 characters. The first character cannot be a number.

Location: Ethernet > Mod Config > WAN Options

Local DNS host address table option added

To have a fallback when the DNS fails to resolve a hostname successfully, you can now create a local DNS host table that supplies a list of host addresses for important or frequently used connections. You can also specify that the table are automatically updated each time the remote DNS succeeds in resolving a name that is in a list.

Overview

You can now create a local DNS table to provide a list of IP addresses for a specific host name when the remote DNS server fails to resolve the host name. If the local DNS table contains the host name for the attempted connection, it provides the list of IP addresses.

You create the DNS table from the terminal server by entering the host names and their IP addresses. A table can contain up to eight entries, with a maximum of 35 IP addresses for each entry. If you specify automatic updating, you only have to enter the first IP address of each host. Any others are added automatically.

Automatic updating replaces the existing address list for a host each time the remote DNS server succeeds in resolving a connection to a host that is in the table. You specify how many of the addresses returned by the remote server can be included in the new list.

On the MAX, the table provides includes additional information for each table entry. The information is in the following two fields, which are updated when the system matches the table entry with a host name that was not found by the remote server:

- # Reads (the number of reads since entry was created)
  This field is updated each time a local name query match is found in the local DNS table.
- Time of Last Read

You can check the list of host names and IP addresses in the table using the termserv command **`show dnstab`**. Figure 13 shows an example of a DNS table on a MAX. Other terminal server commands show individual entries, with a list of IP addresses for the entry.

```
Local DNS Table
Name                      IP Address      # Reads Time of last read
_________________________ _______________ _______ ____________________
1: ""                     ------          ------
2: "server.corp.com."     200.0.0.0       2       Feb 10 10:40:44
3: "boomerang"            221.0.0.0       2       Feb 10  9:13:33
4: ""                     ------          -------
5: ""                     ------          -------
6  ""                     ------          -------
7: ""                     ------          -------
```

Figure 13. Local DNS table example

New terminal server command changes

New *show* and *dnstab* commands have been added to help you view, edit, or make entries in the DNS table.

show commands

- **show ?** displays a list that includes **dnstab** help.
- **show dnstab** displays the local DNS table.
- **show dnstab ?** displays help for the dnstab editor.

dnstab commands

The terminal server **dnstab** command has three variations:

Table 14. dnstab commands

dnstab Command	Description
dnstab	Displays help information about the DNS table.
dnstab show	Displays the local DNS table.
dnstab entry *n*	Displays a list for entry *n* in the local DNS table. The list displayed includes the entry and all the IP addresses stored for that entry up to a maximum number of entries specified in the List Size parameter. If List Attempt=No, no list is displayed.

Parameter Reference

List Attempt

Description: Enables or disables the DNS List Attempt feature. DNS can return multiple addresses for a hostname in response to a DNS query, but it does not include information about availability of those hosts. Users typically attempt to access the first address in the list. If that host is unavailable, the user must try the next host, and so forth. However, if the access attempt occurs automatically as part of immediate services, the physical connection is torn down when the initial connection fails. To avoid tearing down physical links when a host is unavailable, you can use the List Attempt parameter to enable the MAX to try one entry in the DNS list of hosts, and if that connection fails, to try the next entry, and so on, without losing the WAN session. The List Size parameter specifies the maximum number of hosts listed.

Usage: Specify Yes or No. No is the default.

- Yes enables a user to try the next host in the DNS list if the first Telnet login attempt fails, which may prevent the physical connection from being torn down.
- No means the connection fails if the first Telnet attempt is refused. For dial-in users, the physical connection is torn down when the initial connection fails.

Dependencies: If List Attempt = No and Enable Local DNS Table = Yes, the local DNS table has only one entry.

Location: Ethernet>Mod Config>DNS

See Also: List Size, Enable Local DNS Table

List Size

Description: Specifies the maximum number of DNS addresses that are made accessible to terminal server sessions in response to a DNS query. List Size also specifies the maximum number of IP address entries in the Local DNS table.

If List Attempt=Yes and the name server returns an IP address list, the list is copied into the entry in the local DNS table that matches the host name, up to the number of entries you specify in List Size. When a list of IP addresses for an entry is automatically updated, any existing list for that entry is discarded.

Note: The number of IP addresses displayed with the dnstab entry terminal command depends upon the value you set in the List Size parameter.

Usage: Specify a number between 1 and 35. The default is 6.

Example: Following are three possible local DNS table situations:

- You have set List Size=4 and the remote DNS returns 3 addresses, the three addresses replace the entire list of four IP addresses in the local DNS table.
- You have set List Size= 35, and the remote DNS server returns only 4 addresses. The MAX places the four IP addresses in the table and sets the remaining 31 addresses in the list to zero.
- You have just changed the List Size =1. Previously, you had set List Size=10. The next time the table entry for that one IP address is updated, only the first IP address will be retained in the table, all nine others will be set to zero.

Dependencies: This parameter is applicable only when the parameter List Attempt = Yes. A local DNS table is created only if the parameter Enable Local DNS Table= Yes.

Location: Ethernet>Mod Config>DNS

See Also: List Attempt, Enable Local DNS Table

Enable Local DNS Table

Description: Enables the use of a local DNS table that can provide a list of IP addresses for a specific host when the remote DNS server fails to resolve the host name successfully. The local DNS table provides the list of IP addresses only if the host name for the attempted connection matches a host name in the local DNS table.

Usage: Select Enable Local DNS Table=Yes to enable the local DNS table. No disables the feature. No is the default.

Location: Ethernet Profile: Ethernet > Mod Config > DNS

See Also: The *`dnstab entry`* terminal command.

Loc.DNS Tab Auto Update

Description: Enables.or disables automatic updating. When automatic updating is enabled, the list of IP addresses for each entry is replaced with a list from the remote DNS when the remote DNS successfully resolves a connection to a host named in the table.

Usage: Loc.DNS Tab Auto Update=Yes to enable automatic updating of the IP addresses in the local DNS table. No disables automatic updating. No is the default.

When automatic updating is enabled, the list of IP addresses for each entry is replaced with a list from the remote DNS when the remote DNS successfully resolves a connection to a host named on the table.

Dependencies: The Enable Local DNS Table parameter must be set to Yes.

Location: Ethernet Profile: Ethernet > Mod Config > DNS

Configuring the local DNS table

To enable and configure the local DNS table:

1. Display Ethernet Profile: Ethernet > Mod Config > DNS menu.
2. Select a setting for the List Attempt parameter.
3. Specify the list size by setting the List Size parameter.
4. Select Enable Local DNS Table=Yes.
   The default is No.
5. Select a setting for the Loc.DNS Tab Auto Update parameter.

Criteria for valid names in the local DNS table

- Must be unique in the table.
- Must start with an alphabetic character, which may be either upper- or lower-case.
- Must be less than 256 characters
- Names can be local names or fully qualified names that include the domain name.

Periods at the end of names are ignored.

Entering IP addresses in the local DNS table

To enter IP addresses in a local DNS table, you use the DNS table editor from the terminal server. While the editor is in use, the system cannot look up addresses in the table or perform automatic updates. A table *entry* is one of the eight table indexes. It includes the host name, IP address (or addresses), and information fields. To place the initial entries in the table:

1. At the terminal server interface, type **`dnstab edit`**.
   Before you make any entries, the table is empty. The editor initially displays zeros for each of the eight entries in the table. To exit the table editor without making an entry, press Enter.
2. Type an entry number and press Enter.

A warning appears if you type an invalid entry number. If the entry exists, the current name for that entry appears in the prompt.

3 Type the name for the current entry.

If the system accepts the name, it places the name in the table and prompts you for the IP address for the name that you just entered. (For the characteristics of a valid name, see "Criteria for valid names in the local DNS table" on page 53.)

If you enter an invalid name, the system prompts you to enter a valid name.

4 Type the IP address for the entry.

If you enter an address in the wrong format, the system prompts you for the correct format. If your format is correct, the system places the address in the table and the editor prompts you for the next entry.

5 When you are finished making entries, type the letter **`0`** and press Enter when the editor prompts you for another entry.

Editing the local DNS table

To edit the DNS table entries, you access the DNS table editor from the terminal server. While the editor is in use, the system cannot look up addresses in the table or perform automatic updates. A table *entry* is one of the eight table indexes. It includes the host name, IP address (or addresses), and information fields. To edit one or more entries in the local DNS table:

1 At the terminal server interface, type **`dnstab edit`**.

If the table has already been created, the number of the entry last edited appears in the prompt.

2 Type an entry number or press Enter to edit the entry number currently displayed.

A warning appears if you type an invalid entry number. If the entry exists, the current value for that entry appears in the prompt.

3 Replace, accept, or clear the displayed name, as follows:

– To replace the name, type a new name and press Enter.

– To accept the current name, press Enter.

– To clear the name, press the spacebar and then Enter.

If you enter a valid name, the system places it in the table (or leaves it there is you accepted the current name) and prompts you for the corresponding IP address. (For the characteristics of a valid name, see "Criteria for valid names in the local DNS table" on page 53)

If you clear an entry name, all information in all fields for that entry is discarded.

4 Either type a new IP address and press Enter, or leave the current address and just press Enter.

– If you are changing the name of the entry but not the IP address, press Return.

– To change the IP address, type the new IP address

If the address is in the correct format, the system places it in the table and prompts you for another entry.

5 When you are finished making entries, type the letter **`0`** and press Enter when the editor prompts you for another entry.

Deleting an entry from the local DNS table

To delete an entry from the local DNS table:

1 At the terminal server interface, type **`dnstab edit`** to display the table.

2 Type the number of the entry you want to delete and press Return.

3 Press the spacebar and then press Return.

UDP Queue Control

You can now control the size of the SNMP and RIP UDP queues. Additional information is reported in the terminal server show udp listen command.

If SNMP or RIP UDP packets arrive at a rate faster than the MAX can process them, then a backlog builds up. This feature lowers the priority of UDP packets destined for the MAX and allows the user to set the size of the SNMP and RIP queues where UDP packets are stored for deferred processing. Prior to this feature, UDP packets destined for the MAX were processed at the same priority as routed packets; they now have a lower priority. Also, prior to this feature, the SNMP UDP queue had no limit and a flood of packets could cause the MAX to run out of memory.

New Parameters

Queue Depth

Description: The maximum number of unprocessed SNMP requests which the MAX saves. If SNMP requests arrive at a rate faster than they can be processed, then a backlog builds up. This parameter sets the maximum depth of the queue. If the queue fills, further packets destined for it are discarded.

Usage: Enter an integer value from 0 to 1024. If you enter 0, the MAX saves SNMP requests until it runs out of memory. 0 is the default.

> **Note:** Setting Queue Depth to 0 is not recommended. An unlimited queue depth could result in an out-of-memory error on the MAX if it receives a flood of packets on its SNMP port.

Location: Ethernet > Mod Config > SNMP options...

See Also: Rip Queue Depth

Rip Queue Depth

Description: The maximum number of unprocessed RIP requests which the MAX saves. If RIP requests arrive at a rate faster than they can be processed, then a backlog builds up. This parameter sets the maximum depth of the queue. If the queue fills, further packets destined for it are discarded. This limit applies to each RIP socket, so if RIP is running on multiple interfaces, this parameter limits the number of requests stored per interface.

Usage: Enter an integer value from 0 to 1024. If you enter 0, the MAX saves RIP requests until it runs out of memory. 50 is the default.

> **Note:** Setting Queue Depth to 0 is not recommended. An unlimited queue depth could result in an out-of-memory error on the MAX if it receives a flood of packets on its RIP port.

Dependencies: This parameter does not apply if the MAX does not listen to RIP updates.

Location: Ethernet > Mod Config > Route Pref...

See Also: Queue Depth, RIP

Displaying UDP statistics and listen

The show udp listen command now shows these additional parameters:

- InQMax - The maximum number of queued UDP packets on the socket (See Queue Depth and Rip Queue Depth parameters.)
- InQLen - The current number of queued packets on the socket
- InQDrops - The number of packets discarded because it would cause InQLen to exceed InQMax
- Total Rx - The total number of packets received on the socket, including InQDrops

An example follows:

```
ascend% show udp listen

udp:
Socket Local Port InQLen InQMax    InQDrops    Total Rx
     0        1023       0      1           0           0
     1         520       0     50           0         532
     2           7       0     32           0           0
     3         123       0     32           0           0
     4        1022       0    128           0           0
     5         161       0     64           0           0
```

Specifying the metric and preference for offline WAN connections

You can now specify the metric and preference for the MAX to use when a WAN connection is physically down.

User interface changes

The IP Options submenu of the Connection Profile contains two new parameters: DownPreference and DownMetric. The following sections describe each parameter.

DownMetric

Description: This parameter specifies the metric for a route whose associated WAN connection is down.

Usage: Specify an integer. The higher the metric, the less likely that the MAX will use the route. The default metric for online WAN connections is 1. The default metric for offline WAN connections is 7. The metric you specify is in effect only as long as the WAN connection is down.

See Also: DownPreference

DownPreference

Description: This parameter specifies the preference value for a route whose associated WAN connection is down.

Usage: Specify an integer. A higher preference number represents a less desirable route. The default preference for online WAN connections is 60. The default preference for offline WAN connections is 120. The preference you specify is in effect only as long as the WAN connection is down.

Dependencies: Make sure that routes for offline connections have a higher preference number than routes for online connections. The following table lists the factory default values for route preferences.

Table 15. Default route preferences

Route type	Default value
Interface	0
ICMP	30
RIP	100
OSPF ASE	150
OSPF Internal	10
Static	60
Down-Wan	120
Infinite	225

See Also: DownMetric

IPX features

SPX spoofing added for IPX

Overview

This feature spoofs the SPX watchdog so that the WAN connection can remain idle while the application(s) requiring it are idle.

NetWare applications that require a guaranteed packet delivery use the NetWare SPX protocol. This includes applications such as Print Server (PSERVER) and Remote Printer (RPRINTER), as well as Remote Console (RCONSOLE). The client's SPX watchdog monitors the connection with the server while the connection is idle. To monitor the connection, the SPX watchdog sends a query that brings up the WAN connection every 14 seconds while an SPX application is running.

In previous software versions these repeated watchdog packets from the client's SPX watchdog kept the WAN connection up unnecessarily. This features enables the Ascend unit to allow Netware SPX clients to remain logged in without keeping the WAN connection up in times of inactivity.

To do this, the Ascend unit responds to SPX watchdog requests from the LAN with a "fake" SPX-watchdog-reply packet, and drops any SPX-watchdog-alive packets from the LAN, without sending them on to the WAN.

Note: Routers on both ends of the connection must support this feature for it to function.

New Answer profile option for dial-in NetWare clients

A new Answer parameter has been added to optimize IPX routing for dial-in clients.

In previous releases, the MAX always assumed initially that the far end of an incoming IPX connection was another IPX router. After answering the call, the MAX could recognize the caller as a client via the Peer=Dialin setting in the caller's Connection profile. For dial-in Windows 95 clients with no configured profile, this default behavior caused problems: the connection would take more than a minute to establish and then the client could not see NetWare servers on the local network. In this release, the Answer profile also contains a Peer parameter to enable the MAX to treat incoming IPX connections as clients even when configured profiles are not in use.

A new IPX Options submenu in the Answer profile contains the Peer parameter, which enables the MAX to route to dial-in NetWare clients even when the client has no configured profile. The Peer parameter is set to Router by default, which tells the MAX to negotiate inbound IPX calls as if the far end is a router. The Dialin setting tells the MAX to negotiate inbound IPX calls as if the far end is a dial-in NetWare client.

The following listing shows the new Peer parameter as well as other required parameters with example values:

```
Answer
    Profile Reqd=No
    IPX options...
        Peer=Dialin
    PPP options...
        Route IPX=Yes

Mod Config
    Ether options...
        IPX Enet#=cffff123
        IPX Pool#=cf000888
```

Dependencies: The MAX must be configured to answer calls for which no configured profile is found (no Connection profile, Names/Passwords profile, or RADIUS entry). The call may require no authentication, or it may use SecureID passwords. The dial-in client must be running PPP software.

IPX routing must be enabled in the PPP Options submenu of the Answer profile, and the IPX network number of the router's Ethernet interface must be configured in the Ethernet profile.

Dependencies: Dial-in NetWare clients do not have their own IPX network, so to enable the MAX to route to dial-in clients, you must specify an IPX Pool number in the Ethernet profile. The network number you specify must be unique within the entire IPX routing domain of the MAX (the local routing domain as well as all WAN links). This is a "virtual" IPX network reserved for dial-in clients. If the client does not provide its own unique node number, the MAX assigns a unique node number to the client as well. It does not send RIP and SAP advertisements across the connection and ignores RIP and SAP advertisements received from the far end. However, it does respond to RIP and SAP queries received from dial-in clients.

Support for IPX without defining an IPX server

You can now specify a route to a destination IPX network without defining an IPX server in the IPX Routes submenu of the Ethernet configuration profile. Previously, if you specified a route without also specifying an IPX server, the Pipeline would put a NULL entry in the SAP table. This feature modifies this behavior so that no entry is placed in the SAP table.

Interface changes

There are no user interface changes resulting from this feature. The IPX Routes submenu of the remains the same. You can reach an IPX network by entering the Network number (for example, Network=00123456) without specifying the Server Name and Server Type.

SNMP features

Enable and Disable individual modem using SNMP

A new MIB object makes it possible for you to enable or disable an individual modem in a T1 MAX using SNMP.

Changes to the Ascend MIB

You can enable or disable individual modems on MAX units with the T1 interface platform feature using the new MIB object slotMdmItemConfig (1.3.6.1.4.1.529.2.5.1.6). The modem slot card must be enabled in order to enable or disable the modem.

The following was added to the slot modem table for slotMdmEntry.

```
slotMdmItemConfig   OBJECT-TYPE

    SYNTAX        INTEGER {

                  other(1),
                  enable(2),
                  disable(3),
                  disableAndChannel(4)
                  }
    ACCESS        read-write
    STATUS        mandatory
    DESCRIPTION     "The modem configuration state. SETs
                    are allowed only if the corresponding
                     modem slot card is enabled."
          ::= { slotMdmEntry 6 }
```

SNMP write security disabled by default

A new parameter, R/W Comm Enable, whose default is No, disables set commands. Prior to this software release, the default behavior was to allow SNMP set commands.

Enabling SNMP write security

SNMP set commands enable you to load and save an Ascend unit's system configuration using TFTP, and to make changes to the unit's configuration. With this software release, SNMP set commands are not permitted by default.

A new parameter in this feature, R/W Comm Enable, enables you to specify that SNMP set commands are enabled. To enable SNMP set commands:

1 Open the Ethernet > SNMP Options menu.

```
90-B00 Mod Config
 SNMP options...
  Read Comm=public
 >R/W Comm Enable=No
  R/W Comm=N/A
```

2 Set R/W Comm Enable=Yes.

When R/WComm Enable=No, the R/W Comm parameter is N/A.

Note: To use a set command, you must know the read-write community string, even if R/W Comm Enable is set to Yes.

R/W Comm Enable

Description: This parameter enables and disables the use of SNMP set commands.

Usage: Press Enter to select Yes or No.

- Yes enables the use of SNMP set commands. To use a set command, you must know the SNMP read-write community string specified in the R/W Comm parameter.
- No disables the use of set commands.

  No is the default.

SNMP "get" now retrieves MPP session statistics

A new table in the Ascend MIB, mppActiveStatsTable, enables you to use an SNMP get request to obtain MPP the session statistics that appear in the Dyn Stat status display.

Overview

MPP session statistics appear in the Dyn Stat status window. Now, you can use SNMP get requests to query these values.The mppActiveStatsTable, added to the systemStatusGroup of the Ascend MIB (.1.3.6.1.4.1.529.12) to provide the objects necessary for an SNMP get request for session statistics.

Using a get request to obtain MPP session statistics

The user interface changes are within the SNMP get requests. For example, if you use a simple SNMP "walk" utility to perform a "walk" request on the object identifier

```
.1.3.6.1.4.1.529.12.4
```

you will obtain sets of values that correspond to those that appear in the Dyn Stats window for a single MPP session on the LCD display. Each set of values is assigned an MpID.

Note: A "walk" utility is a form of get next request that begins with the zero index. Since the zero index does not exist (the index begins at 1), the utility returns the first available index, which would normally be 1, and continues returning indexes until there are no more available indexes.

Value sets returned by a get request

The values in the table below appear in each set returned by an SNMP "walk" or get request on the mppStatsMpID. For more information on these parameters, see the Reference Guide that came with your documentation.

Value in mppStatsTable	Dyn Stats Window Parameter	Description
mppStatsRemoteName	(*Connection Profile name*)	Connection Profile name set up in the MAX for this connection; shown at top of Dyn Stats window.
mppStatsQuality	Qual	Second line of Dyn Status window Shows the quality of the link. Possible values are: Good, Fair. Marg. Poor., and N/A (link is not online)
mppStatsStartingTimeStamp	(*time*)	The amount of time the link has been active. When the link has been active for more than 96 hours, the duration is reported in days.
mppStatsBandwidth	(*data rate*)	Third line of Dyn Stats window. Shows the current data rate
mppStatsTotalChannels	*n* channels	The number of channels the data rate in mppStatsBandwidth represents.
mppStatsCLU	CLU *n*%	Current line utilization
mppStatsALU	ALU *n*%	Average line utilization.

Changes to the Ascend MIB

A new table, the mppActiveStatsTable, has been added to the SessionStatusGroup in the Ascend MIB to support this feature.

- **`mppActiveStatsTable`** OBJECT-TYPE sessionStatusGroup 4
  SYNTAX SEQUENCE OF MppActiveStatsEntry
  ACCESS not-accessible
  STATUS mandatory
  DESCRIPTION "A list of active MPP session statistics with invalid entries screened out and indexed by mppStatsMpID."
- mppActiveStatsEntry OBJECT-TYPE mppActiveStatsTable 1
  `SYNTAX` MppActiveStatsEntry
  `ACCESS` not-accessible
  `STATUS` mandatory

DESCRIPTION "An entry containing object variables to describe an active MPP session. The variables are those seen in the Dyn Stat area of the LCD display."

- **mppStatsMpID** OBJECT-TYPE mppActiveStatsEntry 1
  SYNTAX `INTEGER`
  ACCESS `read-only`
  STATUS `mandatory`
  DESCRIPTION "The MpID number for this active MPP session entry."
- mppStatsRemoteName OBJECT-TYPE mppActiveStatsEntry 2
  SYNTAX DisplayString
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "The name of the remote user."
- mppStatsQuality OBJECT-TYPE mppActiveStatsEntry 3
  SYNTAX `INTEGER {`
  good(1),
  fair(2),
  marginal(3),
  poor(4),
  na(5)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION Line quality. N/A: No MPP sessions currently active,
  Good: <%1 CRC errors,
  Fair: <%5 CRC errors,
  Marginal: <%10 CRC errors,
  Poor:%10 or > CRC errors"
- **mppStatsBandwidth** OBJECT-TYPE mppActiveStatsEntry 4
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "Total bit rate (Kbps) for the MPP session."
- **mppStatsTotalChannels** OBJECT-TYPE mppActiveStatsEntry 5
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
- **DESCRIPTION** "The total number of channels associated with this MPP session."
- **mppStatsCLU** OBJECT-TYPE mppActiveStatsEntry 6
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "Current percentage of line utilization for transmitted packets during this MPP session."
- **mppStatsALU** OBJECT-TYPE mppActiveStatsEntry 7
  SYNTAX INTEGER

ACCESS read-only
STATUS mandatory
DESCRIPTION "Average percentage of line utilization for transmitted packets during this MPP session."

- **mppStatsStartingTimeStamp** OBJECT-TYPE mppActiveStatsEntry 8
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory

DESCRIPTION "The starting time for this MPP session in seconds since startup."

SNMP can monitor WAN lines and channels

You can now use SNMP to monitor WAN lines without logging into the MAX.

Overview

You can use SNMP to obtain WAN information without logging into the MAX. Each Ascend WAN type is assigned an object ID. These identifiers are the root of the MIB subtree containing WAN-specific inforamation. These subtrees, when appropriate, are described in separate files.

Changes to the Ascend MIB

- wanUseTrunkGroups OBJECT-TYPE wanInfo 20

```
SYNTAX  INTEGER {
do-not-use(1),
use(2)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION  "System wide parameter dictating the use of trunk groups."
```

- wanLineTable OBJECT-TYPE wanInfo 21

```
SYNTAX  SEQUENCE OF WanLineEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION  "The wan line table."
```

- wanLineEntry OBJECT-TYPE wanLineTable 1

```
SYNTAX  WanLineEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION "An entry in the wan line table."
```

- wanLineIfIndex OBJECT-TYPE wanLineEntry 1

```
SYNTAX  INTEGER
```

ACCESS read-only

STATUS mandatory

DESCRIPTION "This value for this object is equal to the value of ifIndex from the Interfaces table of MIB II (RFC 1213)."

- wanLineName OBJECT-TYPE wanLineEntry 2

SYNTAX DisplayString

ACCESS read-only

STATUS mandatory

DESCRIPTION "A textual name of the wanLine as assigned through the menu sytem."

- wanLineType OBJECT-TYPE wanLineEntry 3

SYNTAX OBJECT IDENTIFIER

ACCESS read-only

STATUS mandatory

DESCRIPTION "One of 'wanTypes'."

- wanLineChannels OBJECT-TYPE wanLineEntry 4

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION "The number of ds0 channels supported."

- wanLineState OBJECT-TYPE wanLineEntry 5

SYNTAX INTEGER {

ls-unknown(1),

ls-does-not-exist(2),

ls-disabled(3),

ls-no-physical(4),

ls-no-logical(5),

ls-point-to-point(6),

ls-multipoint-1(7),

ls-multipoint-2(8),

ls-loss-of-sync(9),

ls-yellow-alarm(10),

ls-ais-receive(11),

ls-no-d-channel(12),

ls-active(13),

ls-maintenance(14)

ACCESS read-only

STATUS mandatory

DESCRIPTION "The state of the line."

- wanLineStateString OBJECT-TYPE wanLineEntry 6

SYNTAX DisplayString

ACCESS read-only

STATUS mandatory

DESCRIPTION "A textual representation of the wanLineState as displayed by the menu sytem."

- wanLineActiveChannels OBJECT-TYPE wanLineEntry 7

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION "The number of active ds0 channels of the line."

- wanLineChannelTable OBJECT-TYPE wanInfo 22

S**YNTAX** SEQUENCE OF WanLineChannelEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION "The wan line table."

- wanLineChannelEntry OBJECT-TYPE wanLineChannelTable 1

SYNTAX WanLineChannelEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION "An entry in the wan line table."

- wanLineChannelIfIndex OBJECT-TYPE wanLineChannelEntry 1

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION "This value for this object is equal to the value of ifIndex from the Interfaces table of MIB II (RFC 1213)."

- wanLineChannelIndex OBJECT-TYPE wanLineChannelEntry 2

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION "The ds0 channel number with the line."

- wanLineChannelState OBJECT-TYPE wanLineChannelEntry 3

SYNTAX INTEGER {

bs-unknown(1),

bs-unavailable(2),

bs-unused(3),

bs-out-of-service(4),

bs-nailed-up(5),

bs-held(6),

bs-idle(7),

```
bs-clear-pending(8),
bs-dialing(9),
bs-ringing(10),
bs-connected(11),
bs-signaling(12),
bs-cut-through(13),
bs-current-d(14),
bs-backup-d(15),
bs-maintenance(16),
bs-spc-up(17)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION "The state of the ds0 channel."
```

- wanLineChannelStateString OBJECT-TYPE wanLineChannelEntry 4

```
SYNTAX  DisplayString
ACCESS  read-only
STATUS  mandatory
DESCRIPTION   "A textual representation of the wanLineChannelState as
displayed by the menu sytem."
```

- wanLineChannelErrorCount OBJECT-TYPE wanLineChannelEntry 5

```
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION  "The error count encountered on the channel."
```

- wanLineChannelUsage OBJECT-TYPE wanLineChannelEntry 6

```
  SYNTAX  INTEGER {
  ds0-unused-channel(1),
  ds0-switched-channel(2),
  ds0-cut-through(3),
  ds0-clear-64(4),
  ds0-pri-d-channel(5),
  ds0-nfas-prime-d(6),
  ds0-nfas-sec-d(7),
  ds0-cas-channel(8),
  ds0-spc-channel(9)
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION  "The usage for this ds0 channel."
```

- wanLineChannelTrunkGroup OBJECT-TYPE wanLineChannelEntry 7

```
  SYNTAX INTEGER
```

ACCESS read-only

STATUS mandatory

DESCRIPTION "The trunk group assigned to this channel."

- wanLineChannelPhoneNumber OBJECT-TYPE wanLineChannelEntry 8

SYNTAX DisplayString

ACCESS read-only

STATUS mandatory

DESCRIPTION "The phone number of this channel. This is the number sent to the far end in an inverse multiplexed call when instructing the far end to add more bandwidth. The number should contain the minimum number of digits to identify the channel. If the channel is part of a hunt group, the phone number should be blank."

- wanLineChannelSlot OBJECT-TYPE wanLineChannelEntry 9

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION "A slot number for routing incoming calls associated with the channel. A slot-port number zero means calls arriving on this channel can be routed to any port."

- wanLineChannelPort OBJECT-TYPE wanLineChannelEntry 10

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION "A port number for routing incoming calls associated with the channel. A slot-port number zero means calls arriving on this channel can be routed to any port."

- wanLineChannelNailedState OBJECT-TYPE wanLineChannelEntry 11

SYNTAX INTEGER {

not-applicable(1),

nailed-held(2),

nailed-active(3)

ACCESS read-only

STATUS mandatory

DESCRIPTION "The nailed group associated with the channel."

SNMP can obtain active call status

A new table, the callActiveTable (1.3.6.1.4.1.529.11), has been added to the Ascend MIB. The new table enables you to use SNMP to obtain a listing of all active call-status entries. The information for each call in the listing corresponds to that in the Call Status window, described in the reference guide in your MAX documentation package.

To implement the new table, the following objects have been added to the Ascend MIB:

- callActiveTable OBJECT-TYPE callStatusGroup 16
  SYNTAX SEQUENCE OF CallActiveEntry
  ACCESS not-accessible
  STATUS mandatory
  DESCRIPTION "A list of active call status entries."
- callActiveEntry OBJECT-TYPE callActiveTable 1
  SYNTAX CallActiveEntry
  ACCESS not-accessible
  STATUS mandatory
  DESCRIPTION "An entry containing object variables to describe an active call's status."
- callActiveCallReferenceNum OBJECT-TYPE callActiveEntry 1
  SYNTAX INTEGER (1..'7fffffff'h)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "The unique number identifying the session for which this call is associated."
- callActiveIndex OBJECT-TYPE callActiveEntry 2
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "The index number for this call status entry. Its value ranges from 1 to 'callStatusMaximumEntries'."
- callActiveValidFlag OBJECT-TYPE callActiveEntry 3
  SYNTAX INTEGER {
  invalid(1),
  valid(2)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "valid(2) for all active calls."
- callActiveStartingTimeStamp OBJECT-TYPE callActiveEntry 4
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "The starting time for this call in seconds
  since startup."
- callActiveDataRate OBJECT-TYPE callActiveEntry 5
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "The data rate for ISDN calls or the baud rate for modem calls."
- callActiveSlotNumber OBJECT-TYPE callActiveEntry 6
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory

DESCRIPTION "Identifies the slot of the line being used. Its value ranges between 1 and the value 'slotNumber' in Ascend's slots group. This variable is equivalent to 'slotIndex' in the slot group."

- callActiveSlotLineNumber OBJECT-TYPE callActiveEntry 7
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "Identifies the line for network slots. This variable is equivalent to 'slotItemIndex' in Ascend's slot group."
- callActiveSlotChannelNumber OBJECT-TYPE callActiveEntry 8
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "Identifies the channel for the particular line identified by 'callActiveSlotLineNumber'."
- callActiveModemSlotNumber OBJECT-TYPE callActiveEntry 9
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "Identifies the modem slot on the device. Its value ranges between 1 and the value 'slotNumber' in Ascend's slot group."
- callActiveModemOnSlot OBJECT-TYPE callActiveEntry 10
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "Identifies the particular modem within a modem slot. A value of 0 indicates modems are not involved for this call."
- callActiveIfIndex OBJECT-TYPE callActiveEntry 11
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "The interface index, ranging from 1 to the number of interfaces specified in the MIB-II variable ifNumber. The interface identified by a particular value of this index is the same interface as identified by the same value if ifIndex."
- callActiveSessionIndex OBJECT-TYPE callActiveEntry 12
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "The index of the associated session entry. Value ranges from 1 to 'ssnActiveMaximumSessions'."
- callActiveType OBJECT-TYPE callActiveEntry 13
  SYNTAX INTEGER {
  callOutgoing(1), -- outgoing call
  callIncoming(2) -- incoming calL
  ACCESS read-only

STATUS mandatory
DESCRIPTION "Differenciates between outgoing and incoming calls."

SNMP system reset

You can now reset an Ascend unit using SNMP.

Overview

The new SNMP object `sysReset` enables you to reset an Ascend unit from an SNMP manager. A one-minute timeout after the reset command is issued permits the Ascend unit to confirm the set request before the unit is reset. While the unit is in the process of resetting, it will not respond to other SNMP requests.

Note: You cannot modify the length of the timeout.

Determining whether the Ascend unit has reset

Information held in the Ascend Events Group is erased and its values are initialized when the Ascend unit is reset by software or by toggling the power off and on.
`sysAbsoluteStartupTime` is the time in seconds since January 1, 1990. You can retrieve `sysAbsoluteStartupTime` and compare this value against the previous polls value to determine whether the box has actually been reset by `sysReset`.

Traps generated

The reset process generates the standard traps (see `coldStart` and `warmStart` traps in the *SNMP Supplement*.

Changes to Ascend MIB

A new object has been added to the systemStatusGroup in the Ascend MIB:

```
sysReset            OBJECT-TYPE
    SYNTAX          INTEGER {
                    no-op( 1 ),
                    reset( 2 )
                    }
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION     "The reset takes effect after 1 minute."
    ::= { systemStatusGroup 8 }
```

SNMP can detect concurrent sessions

You can now use SNMP to detect concurrent sessions with a single user.

Overview

A new table, sessionActiveTable, has been added to the Ascend MIB that enables you to detect concurrent sessions with a single user.The MAX must obtain and cache the ssnStatusCallReferenceNumfrom the RADIUS server to be retrieved by the SNMP get request.

Changes to the Ascend MIB

- sessionActiveTable OBJECT-TYPE sessionActiveGroup 3
  STATUS mandatory
  DESCRIPTION "A list of active session entries.
  This table is similar to sessionStatusTable with invalid entries screened out and indexed by:
  ssnActiveCallReferenceNum.
  ssnActiveCallReferenceNum tracks
  ssnStatusCallReferenceNum of
  sessionStatusTable."
- sessionActiveEntry OBJECT-TYPE sessionActiveTable 1
  SYNTAX SessionActiveEntry
  ACCESS not-accessible
  STATUS mandatory
  DESCRIPTION "An entry containing object variables to describe an active session."
- ssnActiveCallReferenceNum OBJECT-TYPE sessionActiveEntry 1
  SYNTAX INTEGER (1..'7fffffff'h)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "A unique number identifying this active session.
  Refer to ssnStatusCallReferenceNum for more information."
- ssnActiveIndex OBJECT-TYPE sessionActiveEntry 2
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "The index number for this session status entry. Its
  value ranges from 1 to 'ssnStatusMaximumSessions'. Refer to ssnStatusIndex for more information."
- ssnActiveValidFlag OBJECT-TYPE sessionActiveEntry 3
  SYNTAX INTEGER {
  invalid(1),
  valid(2)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "All entries will be valid(2). Refer to ssnStatusValidFlag for more information."
- ssnActiveUserName OBJECT-TYPE sessionActiveEntry 4
  SYNTAX DisplayString

ACCESS read-only

STATUS mandatory

DESCRIPTION "The name of the remote user. Refer to ssnStatusUserName for more information."

- ssnActiveUserIPAddress OBJECT-TYPE sessionActiveEntry 5

  SYNTAX IpAddress

  ACCESS read-only

  STATUS mandatory

  DESCRIPTION "The IP address of the remote user. Refer to ssnStatusUserIPAddress for more information."

- ssnActiveUserSubnetMask OBJECT-TYPE sessionActiveEntry 6

  SYNTAX IpAddress

  ACCESS read-only

  STATUS mandatory

  DESCRIPTION "The subnet mask of the remote user. Refer to ssnStatusUserSubnetMask for more information."

- ssnActiveCurrentService OBJECT-TYPE sessionActiveEntry 7

  SYNTAX INTEGER {

  none(1),

  other(2), -- none of the following

  ppp(3), -- Point-To-Point Protocol

  slip(4), -- Serial Line IP

  mpp(5), -- Multichannel PPP

  x25(6), -- X.25

  combinet(7), -- Combinet

  frameRelay(8), -- Frame Relay

  euraw(9),

  euui(10),

  telnet(11), -- telnet

  telnetBinary(12), -- binary telnet

  rawTcp(13), -- raw TCP

  terminalServer(14), -- terminal server

  mp(15) -- Multilink PPP

  ACCESS read-only

  STATUS mandatory

  DESCRIPTION "The current service provided to the remote user. The value none(1) is returned if entry is invalid OR if user dials into the terminal server and is in midst of a login sequence. Refer to ssnStatusCurrentService for more information."

SNMP RFC 1398 Ethernet-like MIB (dot3) support

added

Support has been added to the the MIB for monitoring statistics for Ethernet-like objects, conforming to RFC 1398. This support includes counters for specific types of frames and errors on a per-interface basis.

RFC 1398 is the mib for the Ethernet port in Ascend units.

Set system clock through SNMP

A new object in the Ascend MIB enables you to use SNMP to set the system clock.

The Ascend MIB now includes the following object:

```
sysAbsoluteCurrentTime  OBJECT-TYPE
  SYNTAX         INTEGER (1..'7fffffff'h)
  ACCESS         read-write
  STATUS         mandatory
  DESCRIPTION    "The current system time in seconds since
                 January 1, 1990. Changing this value may
                 result in a change of sysAbsoluteStartupTime
                 and not of sysSecsSinceStartup. The following
                 relationship holds:

                   sysAbsoluteCurrentTime -
                   sysAbsoluteStartupTime = sysSecsSinceStartup."

       ::= { systemStatusGroup 7 }
```

Terminating user sessions using SNMP

You can now terminate a user session using SNMP. You can now set the ssnStatusValidFlag object in the Session Status group of the Ascend MIB. To terminate a user session, set ssnStatusValidFlag to invalid (1).

The new description for `ssnStatusValidFlag` is presented below.

```
ssnStatusValidFlag  OBJECT-TYPE
   SYNTAXINTEGER {
             invalid(1),
             valid(2)
         }
   ACCESSread-write
   STATUSmandatory
   DESCRIPTION"Value indicates whether this entry contains
valid
             information or not. Setting a valid(2) session to
invalid(1)
            causes an effective session termination."
   ::= { sessionStatusEntry 2 }
```

Ascend MIB change supports counters for total and current calls

A change has been made to the Ascend MIB to permit service management (including capacity planning) based upon the number of digital/analog/frame relay calls for each MAX.

How the MIB changes work

This feature changes the Ascend MIB to support counters for incoming and outgoing calls. These counters track the current and total number of digital, analog, and frame relay calls for each MAX. These counters are reset and initialized to zero when the MAX is reset or booted up.

Changes to Ascend MIB

The changes below have been made to the callStatus group to support the call counters.

```
    callCurrentAnalogOutgoing OBJECT-TYPE
   SYNTAXINTEGER
   ACCESSread-only
   STATUSmandatory
   DESCRIPTION"The number of current analog outgoing calls is
returned."
   ::= { callStatusGroup 4 }

    callCurrentAnalogIncoming OBJECT-TYPE
   SYNTAXINTEGER
   ACCESSread-only
   STATUSmandatory
   DESCRIPTION"The number of current analog incoming calls is
returned."
   ::= { callStatusGroup 5 }

    callCurrentDigitalOutgoing OBJECT-TYPE
   SYNTAXINTEGER
   ACCESSread-only
   STATUSmandatory
   DESCRIPTION"The number of current digital outgoing calls is
returned."
   ::= { callStatusGroup 6 }

    callCurrentDigitalIncoming OBJECT-TYPE
   SYNTAXINTEGER
   ACCESSread-only
   STATUSmandatory
   DESCRIPTION"The number of current digital incoming calls is
returned."
   ::= { callStatusGroup 7 }

    callCurrentFROutgoing OBJECT-TYPE
   SYNTAXINTEGER
```

```
   ACCESSread-only
   STATUSmandatory
   DESCRIPTION"The number of current frame relay outgoing calls
is returned."
   ::= { callStatusGroup 8 }

    callCurrentFRIncoming OBJECT-TYPE
   SYNTAXINTEGER
   ACCESSread-only
   STATUSmandatory
   DESCRIPTION"The number of current frame relay incoming calls
is returned."
   ::= { callStatusGroup 9 }

    callTotalAnalogOutgoing OBJECT-TYPE
   SYNTAXINTEGER
   ACCESSread-write
   STATUSmandatory
   DESCRIPTION"The total number of analog outgoing calls since
system bootup or last clear of the variable is returned."
   ::= { callStatusGroup 10 }

    callTotalAnalogIncoming OBJECT-TYPE
   SYNTAXINTEGER
   ACCESSread-write
   STATUSmandatory
   DESCRIPTION"The total number of analog incoming calls since
system bootup or last clear of the variable is returned."
   ::= { callStatusGroup 11 }

    callTotalDigitalOutgoing OBJECT-TYPE
   SYNTAXINTEGER
   ACCESSread-write
   STATUSmandatory
   DESCRIPTION"The total number of digital outgoing calls since
system bootup or last clear of the variable is returned."
   ::= { callStatusGroup 12 }

    callTotalDigitalIncoming OBJECT-TYPE
   SYNTAXINTEGER
   ACCESSread-write
   STATUSmandatory
   DESCRIPTION"The total number of digital incoming calls since
system bootup or last clear of the variable is returned."
   ::= { callStatusGroup 13 }

    callTotalFROutgoing OBJECT-TYPE
   SYNTAXINTEGER
   ACCESSread-write
   STATUSmandatory
   DESCRIPTION"The total number of frame relay outgoing calls
since system bootup or last clear of the variable is returned."
```

```
   ::= { callStatusGroup 14 }

    callTotalFRIncoming OBJECT-TYPE
   SYNTAXINTEGER
   ACCESSread-write
   STATUSmandatory
   DESCRIPTION"The total number of frame relay incoming calls
since system bootup or last clear of the variable is returned."
   ::= { callStatusGroup 15 }
```

Firewall Control Protocol managed by SNMP

SAM firewalls embedded in Ascend products can now be managed through SNMP.

How the Firewall Control Protocol works

The SNMP objects in the sysFcpGroup of the Ascend Enterprise MIB provide a means of enabling and disabling, creating and changing SAM firewalls in the Ascend MIB.

Ascend MIB definitions for the Firewall Control Protocol

```
sysFcpGroup    OBJECT IDENTIFIER ::= { systemStatusGroup 11 }


   sysFcpRuleName    OBJECT-TYPE
   SYNTAX            DisplayString
   ACCESS            read-write
   STATUS            mandatory
   DESCRIPTION    "The name of the firewall rule to be
           enabled or disabled.  This name corresponds with a
           name established when the firewall was created (as
           by the Secure Access Manager)."
   ::= { sysFcpGroup 1 }

   sysFcpExecute    OBJECT-TYPE
   SYNTAX           INTEGER {
                    no-op( 1 ),
                    enb-rule( 2 ),
                    dis-rule( 3 )
           }
   ACCESS           read-write
   STATUS           mandatory
   DESCRIPTION    "Cause a firewall given by the above
           parameters to be affected as requested.
           add-rule causes a dynamic rule to be created;
           del-rule causes a dynamic rule to cease operating."
   ::= { sysFcpGroup 2 }

   sysFcpTimeOut    OBJECT-TYPE
   SYNTAX            INTEGER
```

```
ACCESS            read-write
STATUS            mandatory
DESCRIPTION    "Time, expressed in seconds, during which
        this firewall rule will effect the firewall.  After
        the expiration time, the rule will cease being
        effective exactly as if a del-rule (see
        sysFcpExecute above) had been executed on it.
        Default is 3600 seconds."
::= { sysFcpGroup 3 }

sysFcpExtAddr    OBJECT-TYPE
SYNTAX           IpAddress
ACCESS           read-write
STATUS           mandatory
DESCRIPTION    "Address of entity outside firewall.  This
        value defaults to 0.0.0.0, equivalent to
        a don't care.  May be used when selecting
        firewall to be updated (see sysFcpRoutAddr)."
::= { sysFcpGroup 4 }

sysFcpExtAddrMask    OBJECT-TYPE
SYNTAX               IpAddress
ACCESS               read-write
STATUS               mandatory
DESCRIPTION    "Netmask of entity outside firewall.  This
        value defaults to 255.255.255.255, equivalent
        to a host address if sysFcpExtAddr is non-zero."
::= { sysFcpGroup 5 }

sysFcpExtPort    OBJECT-TYPE
SYNTAX           INTEGER
ACCESS           read-write
STATUS           mandatory
DESCRIPTION    "For external entity, specifies a port number.
        Defaults to 0, equivalent to don't care."
::= { sysFcpGroup 6 }

sysFcpExtPortMax    OBJECT-TYPE
SYNTAX              INTEGER
ACCESS              read-write
STATUS              mandatory
DESCRIPTION    "For external entity, specifies the maximum
        port number of a range of ports.  Defaults to
        0, equivalent to specifying a single port number
        if sysfcpExtPort is nonzero."
::= { sysFcpGroup 7 }

sysFcpIntAddr    OBJECT-TYPE
SYNTAX           IpAddress
ACCESS           read-write
STATUS           mandatory
DESCRIPTION    "Address of entity inside firewall.  This
```

```
        value defaults to 0.0.0.0, equivalent to
        a don't care."
::= { sysFcpGroup 8 }

sysFcpIntAddrMask    OBJECT-TYPE
SYNTAX               IpAddress
ACCESS               read-write
STATUS               mandatory
DESCRIPTION    "Netmask of entity inside firewall.  This
        value defaults to 255.255.255.255, equivalent
        to a host address if sysFcpIntAddr is non-zero."
::= { sysFcpGroup 9 }

sysFcpIntPort    OBJECT-TYPE
SYNTAX           INTEGER
ACCESS           read-write
STATUS           mandatory
DESCRIPTION    "For Internal entity, specifies a port
       number. Defaults to 0, equivalent to don't care."
::= { sysFcpGroup 10 }

sysFcpIntPortMax    OBJECT-TYPE
SYNTAX              INTEGER
ACCESS              read-write
STATUS              mandatory
DESCRIPTION    "For Internal entity, specifies the maximum
        port number of a range of ports.  Defaults to
        0, equivalent to specifying a single port number
        if sysFcpIntPort is nonzero."
::= { sysFcpGroup 11 }

sysFcpRoutAddr    OBJECT-TYPE
SYNTAX            IpAddress
ACCESS            read-write
STATUS            mandatory
DESCRIPTION    "This address may be supplied by the
        management station to choose a firewall for
        alteration. The default for this address is
        0.0.0.0, which would cause the router to use
        sysFcpExtAddr to choose its firewall instead."
::= { sysFcpGroup 12 }

sysFcpAddrOpts    OBJECT-TYPE
SYNTAX            INTEGER
ACCESS            read-write
STATUS            mandatory
DESCRIPTION    "Firewall requests may require additional
        bit-encoded options to determine the firewall's
        new behavior.  This options variable is a mechanism
        to allow those options to be defined at a later
        date."
```

```
::= { sysFcpGroup 13 }
```

SNMP request authentication added

This feature adds proprietary SNMP request authentication, including replay protection, to the existing SNMP v1 implemented in Ascend units, to the Ascend MAX and Pipeline products. This implementation of SNMP request authentication is compatible with standard SNMPv1 practices, and affects the Ascend unit's interpretation of SNMP messages that use it. Previously, Ascend units did not provide authentication of SNMP requests.

Authenticating requests using SNMP

You can use SNMP for security-related operations, such as altering the operational state of the Ascend unit (rebooting, loading configurations, etc.) or firewall configurations. This feature adds an authentication option to existing SNMP v1 that verifies that SNMP requests are only acted upon when they are known to be produced by an authorized system, and then only if they are known to be of recent origin.

This feature is an addition to SNMP v1 already implemented on Ascend MAX and Pipeline units. You can still use SNMP without authentication by using the previous version of the SNMP R/W Comm parameter in Ethernet > Mod Config > SNMP Options.

Authentication Elements

This feature uses 4 elements to authenticate SNMP packets:

- secret authentication key
- data to be authenticated
- time-dependent state variables (for replay protection)
- MD5 hash value calculated over the key, data, and time.

The data, time, and hash values are transmitted with the packet. This allows the management station and Ascend unit to verify that the packet has been produced by an authorized system, and that the packet not been altered or significantly delayed in transmission.

The MD5 hash guarantees a high likelihood that only a system that knows the secret authentication key generated the packet, while the time variables guarantee a high likelihood that an attacker did not collect an authenticated packet and transmit it at a time of its own choosing, after a significant delay.

Community name string changes

Prior to this software release, existing community names on Ascend units were simply ASCII strings with no internal structure. The original SNMPv1 definition of the community string is a string of octets that is compared to a similar string in the receiving SNMP entity. If the string in the packet received exactly matches a community string in the receiving entity, then the packet is considered "authentic".

The defaults for SNMP v1 (without authentication) are:

Ethernet > Mod Config > SNMP Options > Read Comm=public

Ethernet > Mod Config > SNMP Options > R/W Comm=write

You use a new version of the Read/Write community string if you wish to use SNMP authentication, with the format:

Ethernet > Mod_config > SNMP Options > R/W Comm=write|secretkey

This causes the Ascend unit to require SNMP SET REQUEST packets to be authenticated, using "secretkey" as the shared (but not transmitted) secret.

Configuring SNMP Authentication

To configure SNMP authentication, enter the read-write community name in the R/W_comm parameter of the SNMP Options submenu of the Ethernet profile. The read-write community name should have the format

name|*secretkey*

where:

- ***name*** is the name you want to assign to the read-write community name.
- ***secretkey*** is the alphanumeric key used for authentication.
- a vertical bar separates the ***name*** from the ***secretkey***.

Configuring the SNMP manager to use SNMP authentication

To communicate with an Ascend unit that has been configured to use authenticated SNMP, an SNMP management station must construct an SNMP packet using the new format for the Read/Write community string, including the secret key:

name|*secretkey*

If the Ascend unit has been configured to use authenticated SNMP, it will not accept packets from an SNMP management station using the string format without the pipe/vertical bar.

SNMP sysConfigTftpStatus object reports more states

In the Ascend MIB, the read-only sysConfigTftpStatus object in the System Status group previously only reported whether an SNMP-initiated download or upload passed or failed. It now reports a much wider variety of possible states.

```
 sysConfigTftpStatus OBJECT-TYPE
SYNTAX INTEGER {
    ok( 1 ),-- tftp operation succeeded
    notFound( 2 ),-- file not found
    access( 3 ),-- access violation
    noSpace( 4 ),-- no disk space to write file
    badOp( 5 ),-- bad tftp operation
```

```
        exists( 7 ),-- file already exists
        noSuchUser( 8 ),-- no such user
        parameter( 9 ),-- parameter error
        busy( 10 ),-- tftp server cannot handle request
        noResources( 11 ),-- no memory for request
        timeout( 12 ),-- timed out
        unrecoverable( 13 ),-- unrecoverable error
        tooManyRetries( 14 ),-- too many retries
        createFile( 15 ),-- create file
        openFile( 16 ),-- open file
        inProgress( 17 )-- get/put request in progress
    }
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "This variable indicates the status of a save or restore operation
     through tftp."
::= { sysConfigTftp 2 }
```

SNMP now reports reasons for last reset

A new object in the systemStatusGroup in the Ascend MIB and a new trap report the reason for the last system reset. The possible errors are listed below.

sysLastRestartReason object

In the Ascend MIB, the read-only `sysLastRestartReason` object in the System Status group reports the reason the MAX reset.

```
  sysLastRestartReason OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
            "The error code for the previous box restart. The error codes
             are identical to ones obtained via fatal-history from the
             debug monitor screen."
        ::= { systemStatusGroup 12 }
```

sysLastRestartReason trap

The `sysLastRestartReason` trap reports the reason the reason the MAX reset.

```
sysLastRestartReasonTrap        TRAP-TYPE
        ENTERPRISE      ascend
        VARIABLES       { sysLastRestartReason }
        DESCRIPTION     "This trap is sent to all managers having the
                        alarm condition enabled if the
                        sysLastRestartReason is not unknown
```

```
                                (value of 0)."
        ::= 26
```

Definitions of fatal errors

This section describes the fatal errors that the MAX can report.

The following reset is caused when an Assert is placed in the code. This problem can be either hardware or software related. Contact Ascend Technical Support if you experience an FE1 reset.

```
FATAL_ASSERT =              1
```

The following resets are out-of-memory conditions, sometimes termed a memory leak.

```
FATAL_POOLS_NO_BUFFER =     2
FATAL_PROFILE_BAD =         3
FATAL_SWITCH_TYPE_BAD =     4
FATAL_LIF_FATAL =           5
FATAL_LCD_ERROR =           6
FATAL_ISAC_TIMEOUT =        7
```

The following reset is caused by a processor exception error.

```
FATAL_SCC_SPURIOUS_INT =    8

FATAL_EXEC_INVALID_SWITCH = 9
```

The following reset occurs because the MAX tried to allocate a mail message, and there were none left. A reset of this type is usually due to a memory leak.

```
FATAL_EXEC_NO_MAIL_DESC =   10

FATAL_EXEC_NO_MAIL_POOL =   11
FATAL_EXEC_NO_TASK =        12
FATAL_EXEC_NO_TIMER =       13
FATAL_EXEC_NO_TIMER_POOL =  14
FATAL_EXEC_WAIT_IN_CS =     15
FATAL_DSP_DEAD =            16
FATAL_DSP_PROTOCOL_ERROR =  17
FATAL_DSP_INTERNAL_ERROR =  18
FATAL_DSP_LOSS_OF_SYNC =    19
FATAL_DSP_UNUSED =          20
FATAL_DDD_DEAD =            21
FATAL_DDD_PROTOCOL_ERROR =  22
FATAL_X25_BUFFERS =         23
FATAL_X25_INIT =            24
FATAL_X25_STACK =           25
FATAL_ZERO_MEMALLOC =       27
FATAL_NEG_MEMALLOC =        28
```

The following reset is caused by a software loop.

```
FATAL_TASK_LOOP =           29
```

```
FATAL_MEMCPY_TOO_LARGE =    30
FATAL_MEMCPY_NO_MAGIC =     31
FATAL_MEMCPY_WRONG_MAGIC =  32
FATAL_MEMCPY_BAD_START =    33
FATAL_IDEC_TIMEOUT =        34
FATAL_EXEC_RESTRICTED =     35

FATAL_STACK_OVERFLOW =      36
FATAL_MBUF_PANIC =          38
FATAL_PROTECTION_FAULT =    40
```

The following entry is logged to the fatal-error table when the MAX has been manually reset, either in Diagnostic mode (with the RESET or NVRAMCLEAR commands), through the user interface, or through MIF.

Instead of a standard stack backtrace, the message includes the active security-profile index. The numbering is one-based, with 0 indicating an unknown security profile. On the MAX the Default profile is number 1, and the Full Access profile is number 9.

This reset is logged immediately before the MAX goes down.

```
FATAL_OPERATOR_RESET =      99
```

As a complement to entry 99, the following entry is logged as the MAX is coming up. For a normal, manual reset, you should see a fatal error 99 followed by a fatal error 100.

```
FATAL_SYSTEM_UP =           100
```

Tunneling features

Maximum number of ATMP Tunnel sessions can be set

You can now specify the maximum number of active ATMP tunnel sessions allowed through a unit configured as an ATMP Home Agent. Previously, there was no way to limit ATMP sessions. With ATMP enabled, it was possible for all active sessions to be ATMP sessions.

Max ATMP Tunnels

Description: Defines the maximum number of active ATMP sessions for units configured as an ATMP Home agent.

Changes take effect after the Connection Profile is saved, the connection is cleared, then reestablished.

Usage: Press Enter to open the text field. Type the number of simultaneous ATMP sessions you want to allow through this ATMP Gateway. The default, 0 (zero), disables the parameter.

Dependencies: Applies only to units configured as ATMP Home agents.

Location: Ethernet > Connections > *any profil*e > Session Options menu.

See Also: ATMP Mode, ATMP Gateway.

ATMP inactivity timer

You can now configure a timer for ATMP tunnels, indicating the number of minutes the Ascend unit maintains an idle tunnel before disconnecting it.

Overview

ATMP tunnels between an ATMP foreign agent and an ATMP home agent are disconnected when the foreign agent detects the client, for which the tunnel was created, disconnects. However, if you reset a foreign agent, the home agents of any existing tunnels get no notification that the tunnels should be disconnected.

When an Ascend unit, acting as an ATMP foreign agent, restarts, tunnels that were established to any home agent are not normally cleared, because the home agent is never informed that the mobile clients are no longer connected. The unused tunnels continue to consume resources on the home agent until the Ascend unit acting as the home agent, is restarted. To enable the home agent to reclaim the resources held by unused tunnels, you can configure the Idle limit parameter to indicate how long a home agent maintains an idle tunnel before disconnecting it.

If you set the ATMP inactivity timer, it will not affect any other idle timers you might have configured on the MAX. The MAX can apply more than one idle timer to a particular connection.

Idle limit

Description: Specifies the number of minutes the Ascend unit, configured as an ATMP home agent, maintains an idle ATMP tunnel before it disconnects the tunnel.

Changes you make to Idle limit are enabled for any new tunnels. Existing tunnels are not affected.

Usage: Specify the number of minutes the home agent should maintain any ATMP tunnel before disconnecting it.

Disable the ATMP inactivity timer by setting it to 0. This is the default.

Example: `Idle limit=15`

Dependencies: Idle limit is not applicable if you set the Ethernet > Mod Config > ATMP > ATMP Mode parameter to either Disabled or Foreign.

Location: Ethernet > Mod Config > ATMP

See Also: ATMP Mode, Type, Passwd, SAP Reply, UDP Port

Administration features

Data rates reported to syslog

Two optional fields in the call-cleared Syslog posting show the transmit and receive data rates of the call. If the data rate is known, it is reported in bits per second after the "p" progress code, using the following identifiers:

- **s** (shows the call's transmit rate)
- **r** (shows the call's receive rate)

For example, this example output shows two messages reporting the data rates in syslog.

```
...  ASCEND: call 1 AN slot 3 port 1 VOICE
...  ASCEND: call 2 AN slot 9 port 1 56KR
...  ASCEND: call 2 CL 0K  u=Torning c=185 p=60 s=64000 r=64000
...  ASCEND: slot 9 port 1, line 1, channel 1, Call Disconnected
...  ASCEND: call 1 CL 0K   c=20 p=40 s=31200 r=33600
...  ASCEND: call 3 AN slot 3 port 2 VOICE
...  ASCEND: call 3 CL 0K   c=185 p=31
...  ASCEND: slot 3 port 2, line 1, channel 2, Call Disconnected
```

If the data rate is not known it is omitted, as shown in the last three lines of the example Syslog output immediately above, where a call was placed to the Ascend device but no connection was made.The practice of omitting the data rate where not relevant is in accord with the handling of other fields in the same message.

Receive and transmit rates now reported

The terminal server show users command, RADIUS, and SNMP now report and display the receive and transmit data rates for user sessions. Previously, the transmit rate was not recorded or reported separately from the receive rate.

Displaying user session statistics

At the terminal server prompt type the following to display user session status:

show user

The new show users format is:

```
I Session    Line:  Slot: Tx    Rx    Service    Host          User
O ID         Chan   Port  Data  Rate  Type[mpID] Address       Name
O 231849873  1:1    1:9   56K   56K   MPP[1]     192.168.68.2  jdoe
I 231849874  1:1    3:1   28800 33600 Termsrv     N/A           Modem
3:1
```

at the terminal server prompt.

Previously, the Data Rate column displayed the bearer capacity or modem speed as appropriate to the session type. The Data Rate column has been removed from the display, and two new columns have been added:

- Rx (Receive) Rate
  Shows the receive data rate in bits per second.

- Tx (Transmit) Rate

  Shows the transmit data rate in bits per second.

Note: This feature does not apply to LAN modem cards. These cards do not support any asymmetric data rate connection types, so the transmit rate is the same as the receive rate.

Changes to the Ascend MIB

You can use SNMP to obtain the same received and transmit data rate statistics as shown by the show users command. To accommodate this, three new objects have been added to the Ascend MIB, and three objects have changed their meaning:

- callStatusXmitRate (callStatusEntry 14) (1.3.6.1.4.1.529.11.2.1.14

  callStatusXmitRate is similar to callStatusDataRate, but provides the transmit data rate for an ISDN call or the baud rate for a modem call. callStatusDataRate now indicates the receive data rate for an ISDN call or the baud rate for a modem call.
- callActiveXmitRate (callActiveEntry 14) (1.3.6.1.4.1.529.11.16.1.14)

  The callActiveTable contains a list of active call status entries. callActiveXmitRRate is a entry in callActiveEntry, which contains object variables that describe an active call's status. callXmitRate provides the transmit data rate for an ISDN call or the baud rate for a modem call.
- eventXmitRate (eventEntry 23) (1.3.6.1.4.1.529.10.4.1.23)

  eventXmitRate is similar to eventDataRate, an object variable in an eventEntry object in the eventTable. eventXmitRate provides the transmit data rate for an ISDN call or the baud rate for a modem call.

Syslog enhancements

Syslog has been enhanced to report more information about calls. These enhancements include detailed information about authenticated calls when they disconnect and the Dialed Number Identification Service (DNIS) and Calling Line ID (CLID) for each call.

Call information forwarded to syslog when call terminates

The MAX now sends the syslog daemon information about an authenticated connection when the call terminates. To enable this feature, a new parameter, LogCallInfo, has been added to the Ethernet > Mod Config > Log submenu.

When LogCallInfo is set to EndOfCall, the MAX sends a one-line syslog message to the syslog host when an authenticated call terminates. The message contains the following information:

- Connection information
  - station-name
  - calling phone number
  - called phone number
  - encapsulation protocol
  - datarate (in bps)

  - progress code/disconnect reason
- Authentication information
  - time spent before authenticating (in seconds)
  - bytes/packets received during authentication
  - bytes/packets sent during authentication session
- Session information
  - time spent in session (in seconds)
  - bytes/packets received during session
  - bytes/packets sent during session

For example:

```
"Conn=("cvonk-p50" 5105558581->? PPP 56000 60/185) \
Auth=(3 347/12 332/13) \
Sess=(1 643/18 644/19), Terminated"
```

If some of the information is not available, that field is displayed as either a question-mark (for strings) or a zero (for numerals).

Note: This feature is intended for diagnostic support. It uses User Datagram Protocol (UDP), which provides no guaranteed delivery, so it should not be used for billing purposes.

Parameter reference

This section describes the new parameter to support this feature.

LogCallInfo

Description: Specifies whether the MAX should send connection, authentication and session information about authenticated calls to Syslog when the call terminates. The MAX reports the following information:

- Connection information
  - station-name
  - calling phone number
  - called phone number
  - encapsulation protocol
  - datarate (in bps)
  - progress code/disconnect reason
- Authentication information
  - time spent before authenticating (in seconds)
  - bytes/packets received during authentication
  - bytes/packets sent during authentication session
- Session information

- time spent in session (in seconds)
- bytes/packets received during session
- bytes/packets sent during session

Usage: Specify one of the following values. The Default is None:

- None specifies that the MAX does not report any information about authenticated calls to Syslog when the call disconnects.
- EndOfCall specifies that the MAX reports connection, authentication and session information about authenticated calls when they disconnect.

Dependencies: Keep in mind the following additional infomration:

- LogCallInfo does not apply if Syslog is not enabled.

Location: Ethernet > Mod Config > Log

See Also: Syslog, Log Host, Log Port, Log Facility

Syslog now reports DNIS and CLID information

The MAX did not previously report Calling Line ID (CLID) and Dialed Number Identification Service (DNIS) to Syslog.

Syslog messages that were previously displayed like this:

```
[1/1/1/5] [MBID 2] Incoming Call
```

will now appear as follows, whenever possible:

```
[1/1/1/5] [MBID 2; 5107891212->7242] Incoming Call
```

This format indicates that the Caller ID (5107891212 in the example above) is calling the number 7242.

Syslog messages generated when Security profile activated

This feature enables you to detect and handle unauthorized Telnet or serial-port sessions with the MAX. When a user activates a Security profile, the MAX generates a Syslog message notifying you that the event occurred. Together with companion EOIs 4099 and 4101, this feature forms a security enhancement package, FID 37.

New Syslog messages

A user can activate a Security profile in a Telnet session or a serial-line COM port session by selecting the Security profile and specifying the proper password. When a user activates a Security profile, the new Syslog messages show the name of the Security profile, the IP address of the Telnet client or the COM port number, and the local IP address.

The EventSyslog message has one of these formats:

```
^DP(assword)ASCEND: "<profile_name>" ... for <remote_IP> on <local_IP>
ASCEND: "<profile_name>" ... from <COM_port> on <local_IP>
```

- The <profile_name> argument specifies the name of the activated Security profile.
- The <remote_IP> argument specifies the IP address of the Telnet client.
- The <local_IP> argument specifies the local IP address of the MAX.
- The <COM_port> argument specifies the COM port number for the session.

On system login, the MAX does not generate a Syslog message for the Default Security profile; for all events other than system login, the MAX generates a Syslog message for the Default Security profile. If Syslog is enabled, messages at LEVEL_NOTICE appear when a user activates a Security profile and the MAX accepts the Security profile password.

These two messages signal that a Telnet client has enabled a Security profile:

```
Jan 10 10:05:17 eng-lab-141 ASCEND: "Full Access" security profile
enabled for 206.65.212.9 on 192.168.6.141.
Jan 10 10:07:26 eng-lab-141 ASCEND: "Default" security profile enabled
for 206.65.212.23 on 192.168.6.141.
```

This message signals that a COM port user has enabled the Full Access profile:

```
Jan 10 10:03:52 eng-lab-141 ASCEND: "Full Access" security profile
enabled from com port 0 on 192.168.6.141.
```

Configure port for Syslog messages

To allow you more flexibility in controlling ports in a Syslog host, you can now specify the port at which a remote Syslog host listens for Syslog messages from an Ascend unit. This feature enables you to run multiple copies of the syslog daemon on the Syslog host, with Ascend units sending syslog messages to different ports.

Overview

You can now specify the port at which a remote host listens for syslog messages from an Ascend unit. Syslog messages include warning, notice, and CDR (Call Data Reporting) records from the local system logs that are sent to the Syslog host. The Syslog host is the station to which the Ascend unit sends system log messages, and the Log Port is the port on the Syslog host at which the host listens for these messages.

Previously, the Syslog host was always assumed to listen at a well-known port (port 514). You could not specify a different port.

Configuring the Log Port

1 Open the Ethernet > Mod Config menu.

```
90-C00 Mod Config
 Log...
  Syslog=Yes
  Log Host=206.65.212.205
```

```
>Log Port=514
Log Facility=Local0
```

2. Make sure that Syslog is enabled and a Log Host IP address is specified.
3. Select Log Port and type the port number at which you want the Syslog host to listen for messages from this Ascend unit.

   The default port is port 514.
4. Close the Mod Config menu and save your changes.

Defender authentication enhancements

Syslog messages are now logged when a telnet client logs in and when a Security profile is activated. This feature helps detect and control unauthorized telnet sessions and possible security breaches in the MAX.

When a telnet "accept" takes place, a syslog message now is logged showing the IP address of the telnet client. When a caller again tries to make a connection, the router begins the process of connecting to hosts until it either succeeds or fails after trying one authentication host after another until the entire list of hosts has been processed. This allows the router to authenticate a subsequent user when an authentication host becomes available.

Alternate authentication host configuration

To implement this change, the Defender authentication method now supports addresses for more than one authentication host. In previous releases, the Auth Host #2 and #3 parameters were N/A when Auth was set to Defender. In the current release, the Defender configuration could include an alternate authentication host; for example:

```
Mod Config
    Auth...
        Auth Host #1=137.175.80.62
        Auth Host #2=0.0.0.0
        Auth Host #3=137.175.80.24
        Auth Port=2626
        Auth Src Port=0
        Auth Timeout=30
        Auth Key=
```

There is no problem with "skipping" an authentication host, such as the null address shown for Auth Host #2.

Syslog messages

This release also supports a set of syslog messages reporting the status of the Defender authentication subsystem. The new syslog messages are reported with "LOG_DEFAULT" and "LEVEL_INFO" priorities; for example:

```
Nov 14 15:59:34 137.175.85.20 ASCEND: AuthHost 137.175.81.24 Acti-
vated
Nov 14 15:51:10 137.175.85.20 ASCEND: AuthHost 137.175.81.24 Fails
auth
Nov 14 15:51:10 137.175.85.20 ASCEND: AuthHost 137.175.80.24
```

```
Refuses connect
Nov 14 16:03:05 137.175.85.20 ASCEND: AuthHost 137.175.81.24 Closed
connection
Nov 14 16:05:59 137.175.85.20 ASCEND: AuthHost 137.175.81.24
Address Changed
Nov 14 16:06:31 137.175.85.20 ASCEND: AuthHost 137.175.81.24 New
Authmethod
```

All Defender syslog messages report the standard Ascend header plus Defender-formatted detail: "AuthHost xx.yy.zz.aa StatusX" where "StatusX" has the following identities and meanings:

- Activated

  A Defender Host has been found, with a successful connection established. This state is reported when an authentication session is active and ready to authenticate.
- Fails auth

  A Defender Host has been found, but the router and the Defender authentication server do not agree on their mutual authentication key.
- Refuses connect

  The host either is not responding at all, or has no active Defender Server running.
- Closed connection

  An active Defender authentication server has ended its connection. This would reflect either a failure of the server software or explicit request for the server to shutdown by an administrator.
- Address Changed

  The router administrator has elected to change the IP address of the active authentication server, its port number, or the authentication key. This forces the Defender authentication subsystem to close its connection with the active server and start searching for a new one.
- New Authmethod

  The router administrator has changed the authentication method from "DEFENDER" to something else, causing the Defender authentication subsystem to break an active connection.

RLOGIN enhanced -l option

The “-l” option of the rlogin feature has been modified. Rlogin functionality has not been changed. However, the “-l” option can now be specified either before or after the hostname.

Starting an Rlogin session

From the command line, the user can enter the RLOGIN command in either of the following formats:

- `rlogin [-e<char> -l<username>] <host-name>`
- `rlogin [-e<char>] <host-name> [-l<username>]`

If DNS is configured in the MAX Ethernet Profile, you can specify the host’s name, for example:

```
rlogin myhost
```

If DNS is not configured in the Ethernet Profile, you must type the host's IP address, for example:

```
rlogin 10.2.2.2
```

The -e option is used to set the escape character to the specified character. The default for the escape character is a tilde (~).

The -l option is used to specify a login name other than the name used to log into the terminal server. However, the -l option does not apply to logins authenticated by RADIUS.

To close the Rlogin session, the user must type a carriage return followed by the escape character. For example:

```
<CR>~
```

TFTP checks for compatibility of downloaded files

With this release, the Ascend unit compares the software to be TFTP-uploaded to the currently loaded software. If the platform or network interface does not match, the Ascend unit aborts the upload and displays information about why the abort occurred. The MAX will bypass this check if you use the TFTP command with the -f flag.

This feature protects you from unknowingly uploading software that is incompatible with your Ascend unit. Previously, you were able to upload any software to any Ascend unit. If you uploaded an incompatible software load, the upload would fail and revert to the previously-loaded software, but you received no indication of why the upload failed.

This check is initiated by the currently-loaded software. If your Ascend unit is using a version of software with this feature and you attempt to load an older version of software that does not have this feature, the upload will be aborted because the older software has no platform identifiers that the currently-loaded software uses to validate compatibility, In this case, you'll need to use TFTP with the -f flag to have the Ascend unit upload the older software without performing the compatibility check.

Examples

In the following example, a user attempts to use TFTP to upload a MAX 1800 software load (b.m18) to a MAX 4000 running t.m40:

1 From the vt100 interface, accesses the diagnostics monitor.

2 Enters the following command:

```
tloadcode tftpserver.ascend.com b.m18
```

3 The MAX 4000 displays the following information to the screen:

```
saving config to flash

.....................................
.
loading code from tftpserver.ascend.com
file /tftpboot/b.m18...
thin load:
```

```
This load appears to be for another platform.
This load appears not to support your network interface
Download aborted. Use 'tloadcode -f' to force.
```

The MAX 4000 has compared the uploading file, b.m18 to its currently-loaded file, t.m40. This informational messages indicate that the user attempted to load an incompatible platform and an incompatible network interface.

In the following example, a user attempts to use TFTP to upload an old version of software (without this feature) to a MAX 4000 that uses this feature:

1 From the vt100 interface, accesses the diagnostics monitor.

2 Enters the following command:

```
tloadcode tftpserver.ascend.com t.m140
```

3 The MAX 4000 displays the following information to the screen:

```
saving config to flash

.....................................
.
loading code from tftpserver.ascend.com
file /tftpboot/t.m40...
thin load:
This load has no platform identifier. Proceed with caution.
Download aborted. Use 'tloadcode -f' to force.
```

In the previous example, the user decides that he or she requires the older version and forces the upload. The following messages are displayed:

1 User enters the following command

```
tloadcode -f tftpserver.ascend.com t.m140
```

2 The MAX 4000 displays the following messages:

```
Download forced by user...
.....................................................
.....................................................
.....................................................
>
```

Appletalk features

AppleTalk routing added

An Ascend MAX or Pipeline can now function as an AppleTalk internet router, providing routing functions for AppleTalk nodes (Macintosh workstations or Apple printers) that are connected to the Ascend unit over AppleTalk Remote Access (ARA), Ethernet, or a WAN. Previously, Ascend units used bridging to provide AppleTalk connectivity.

To perform AppleTalk routing with your MAX or Pipeline, you should be sure the software load or Pipeline AppleTalk software you download contains the AppleTalk routing feature.

The new feature is designed for routing LAN traffic over synchronous PPP, MP, MPP, and Frame Relay WAN links. This implementation does not provide support for single-station dial-in over asynchronous PPP links such as modems. Currently, AppleTalk Remote Access (ARA) is the only way to dial into a remote MAX or Pipeline from a computer running the Mac OS.

Caution: Due to memory constraints, the AppleTalk software load disables firewalls on the Pipeline 50, Pipeline 75, and Pipeline 130.

Protocols implemented

MAX and Pipeline Appletalk routing support the following protocols:

- Datagram Delivery Protocol (DDP)
- Routing Table Maintenance Protocol (RTMP)
- AppleTalk Echo (AEP)
- Zone Information Protocol (ZIP)
- Name Binding Protocol (NBP)
- ATCP AppleTalk Control Protocol (ATCP—for router-to-router applications)

(For configuration instructions, see “Configuring AppleTalk routing” on page 99.)

Understanding network ranges and AppleTalk zones

AppleTalk routers provide for the configuration of network numbers and ranges and AppleTalk zones. Network numbers are assigned to network segments, and must be unique within the internetwork. A network range is a range of network numbers specified in the port descriptor of the router port and transmitted through RTMP to the other nodes of the network. Each of the numbers within a network range can represent up to 253 devices.

A zone is a multicast address containing an arbitrary subset of the AppleTalk nodes in an internet. Each node belongs to only one zone, but a particular extended network can contain nodes belonging to any number of zones. Zones provide departmental or other groupings of network entities that a user can easily understand.

In the Ascend AppleTalk router, zone names are not case sensitive. However, some routers regard zone names as case sensitive, and you should be consistent in spelling zone names when you configure multiple connections or routers. Although AppleTalk permits the use of spaces in zone names, it does not consider an underscore to be the same as a space. Since some routers do equate the underscore and the space, or do not recognize a space as a valid character, it is advisable to use only the underscore in a network with routers other than Ascend routers.

Extended and nonextended AppleTalk networks

AppleTalk uses two types of subnetworks: extended and nonextended. Nonextended networks theoretically allow up to 254 nodes. A nonextended network has one network number (not a range) and one zone. Examples of nonextended networks are LocalTalk and ARA dial-up networks.

An extended network is a group of nonextended networks on the same physical data link. It contains a range of network numbers, with each network in the range supporting up to 253 devices. EtherTalk and TokenTalk are examples of extended networks.

Seed routers

At least one router on a network, called a seed router, must have the network-number range in its port description. Other routers on the network can have a network range of 0, which means that they acquire the network-number range from RTMP packets sent by the seed router. To prevent conflicts, all seed routers on the same network must have the same value for the start and end of the network-number range.

Figure 1 shows a network with three routers and three zones configured. Each zone has a range of network numbers.

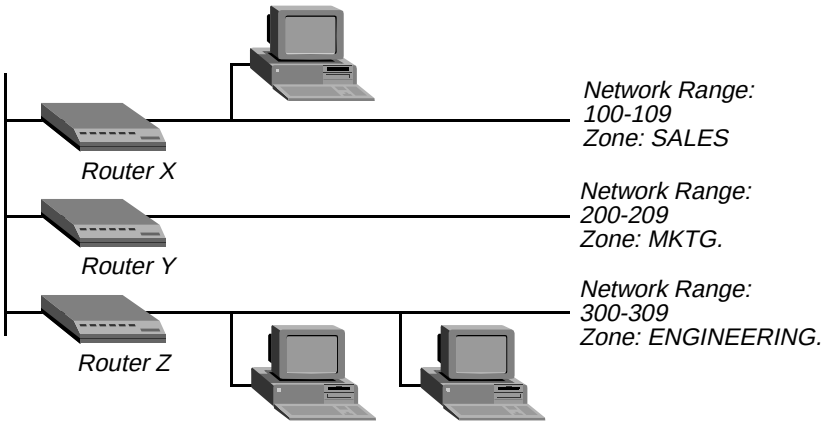


Figure 16. AppleTalk LAN

How AppleTalk works

Figure 2 shows a typical AppleTalk connection. The AppleTalk workstation is part of an Ethernet LAN connected to a Pipeline 75, which has a synchronous PPP WAN connection to a MAX 4000. One of the MAX 4000 ports is on a LAN that includes an Apple Laserwriter printer. Following is a brief, generalized description of how the workstation sends a file to the Laserwriter for printing:

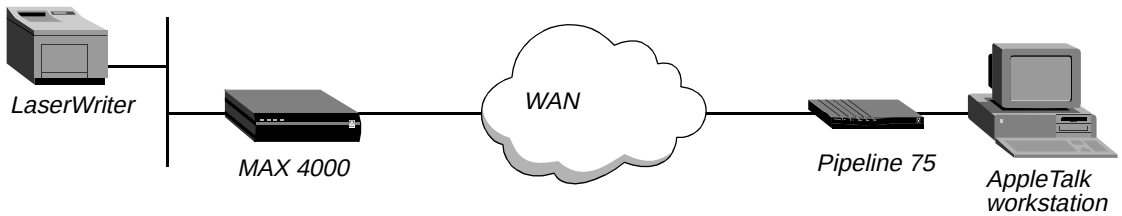


Figure 17. Routed connection

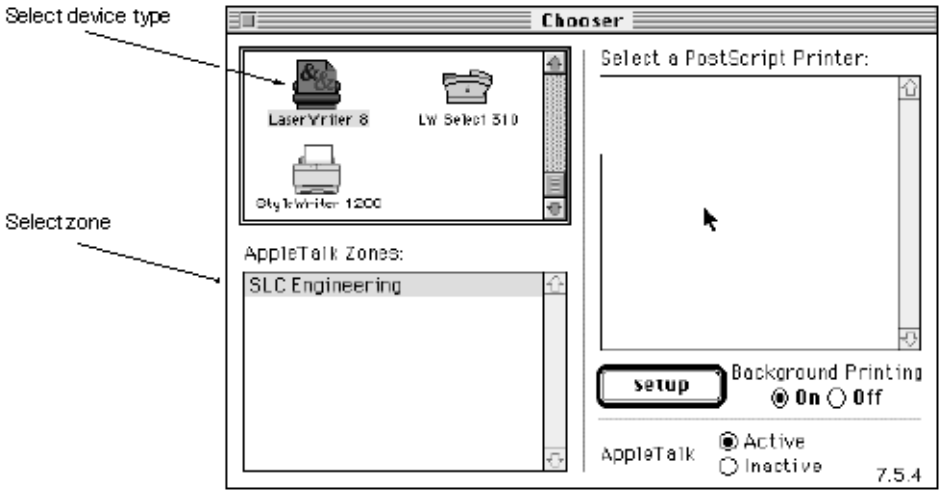
1 The AppleTalk workstation user opens the Macintosh Chooser.

The screen displays the network zones specified by the Connection profile stored in the Pipeline. (The first time a user opens the chooser, only the local Ethernet zones appear. That is, the WAN zone is the same as the local Ethernet zone.)

2 The Pipeline places the call and negotiates the WAN connection with the MAX 4000.

3 The workstation sends a ZIP Query to obtain an updated zone list from the MAX 4000, and the MAX returns the updated zone list. The new list, which might contain different zones from the initial zone list, replaces the initial list in the display and updates the Connection profile in the Pipeline.

4 The user selects a zone and a specific device in the Chooser.



5 The workstation sends a Name Binding Protocol (NBP) Broadcast Request to the Pipeline, which checks its Zone Information Table to identify the subnetwork in which that printer is located, and then sends the request to the MAX via the port configured in the Connection Profile.

6 The MAX determines the port to which the printer's subnetwork is attached, and looks up the printer by searching the multicast address assigned to the zone specified by the Pipeline.

7 All devices in the zone detect and process the NBP-lookup packet.

8 The selected printer obtains the sender's address from the lookup packet (sent by the workstation and forwarded by the routers), and sends the reply through the routers to the workstation.

9 The user sends the print job to the printer.

10 When the print job is complete and no data packets are passing through the connection, the MAX and the Pipeline continue to pass routing information until the idle timeout closes the connection. RTMP and ZIP packets do not reset the idle timer, but any other routeable packet to the network number or zone name specified for this connection does reset the timer.

After the link is dropped, the Pipeline retains in memory the last zone list displayed. If the workstation user opens the Chooser again, the list reappears and the process can begin again.

When to use AppleTalk routing

Use AppleTalk routing to connect two or more networks that have AppleTalk nodes such as Mac OS computers or Apple printers. Although you could use bridging, routing gives you more control over calls, reduces broadcast and multicast traffic over the WAN, and provides startup information to Appletalk nodes.

Call control

Bridging does not provide the call control that routing does, because bridging cannot detect NBP, RTMP, or ZIP packets, all of which can keep a connection open unnecessarily. If you use bridging with a filter to remove these packets from your WAN traffic, the routing information becomes incorrect. A router does not have to bring up a line until it receives a data packet destined for the network number or zone name specified for a remote connection.

NBP packets

The Name Binding Protocol (NBP) drives call placement for the AppleTalk router. NBP is the protocol that enables AppleTalk users to issue a query (in the Chooser) for a type of service and receive specific server or printer names in response. For example, if a user selects LaserWriter in the Chooser and then selects a zone that is on a remote MAX, an NBP query goes to the router destined for the remote network that contains the selected zone. The router brings up a WAN connection in response to the NBP query, and places calls to connections that have AppleTalk routing enabled. The local AppleTalk router and the remote router exchange routing information until the idle timeout closes the connection.

RTMP packets

The AppleTalk router sends Routing Table Maintenance Protocol (RTMP) packets every 10 seconds. If you use bridging for AppleTalk, these packets keep a WAN link up for no other reason than to keep the RTMP information fresh across the WAN. If you add a filter to block RTMP packets, the zone and routing information age and become incorrect. But since RTMP information changes infrequently, it is not necessary to keep the WAN link up all the time. You really need only keep the information the same for the two ends of the WAN.

Once the connection is dropped (due to idle timeout), the Ascend AppleTalk router spoofs the RTMP and ZIP information on the Ethernet network until an event (such as a routeable packet to the network number or zone specified for this connection) requires bringing the WAN link up again. Until the event occurs, the user continues to the same zones see in the Chooser that were established by the connection, but the connection is not active.

RTMP (Routing Table Maintenance Protocol) and ZIP (Zone Information Packets) do not cause a call to be placed, and do not reset the idle timer. Any routeable packet to the network number or zone name specified for a connection brings the link back up.

Dropped connections in routed and bridged AppleTalk

If you filter RTMP packets in a bridged connection, you lose the AppleTalk network context when the connection closes at the end of a call. If another device comes onto the network while the first call is down, it might be assigned the same address as the device at the local end of the dropped call. Addressing inconsistency and network errors can result.

Because the Ascend AppleTalk routing implementation spoofs the RTMP and ZIP information on the Ethernet network, it can drop a call, freeing up the connection, without causing addressing inconsistencies.

Spoofing does not apply when an AppleTalk Filing Protocol volume is mounted on a workstation (selected in the Chooser). In this instance the server and workstation continue to exchange packets with numbers incremented serially, which would be very difficult to emulate or spoof.

Reducing broadcast and multicast traffic

Because AppleTalk uses multicast and broadcast addresses extensively, routing AppleTalk can greatly improve the efficiency of a LAN or WAN. By using AppleTalk zones to segment traffic, you can significantly reduce the amount of broadcast and multicast traffic on a LAN or WAN. When you set up a router for the first time, you identify the cable range (network number) for the subnetwork segment and one or more zones.

For example, when a user on a network without a router selects a device in the Chooser, the MAC OS computer sends out an Name Binding Protocol (NBP) Lookup as a broadcast packet. Since a bridge forwards all broadcast traffic, all devices on the network receive the Lookup packet. A router can significantly reduce AppleTalk traffic over the WAN because it does not forward broadcast traffic from one subnetwork to another, but stops it at the subnetwork port of the router.

Zone multicasting is intended to prevent any node not in the destination zone for the lookup from receiving the lookup packet. Any AppleTalk node responds only to NBP lookups for that node's zone name. In the example above, a router would convert the Broadcast Request packet generated by the Lookup request to a Forward Request packet for each network that contains nodes in the target zone specified by the Lookup request.

Routing vs. bridging multicast and broadcast traffic.

A bridge can filter directed traffic (traffic between two specific nodes) but cannot filter broadcast or multicast traffic, since there isn't a specific port that can be assigned to a multicast or broadcast address. This means that although filters used with bridging can reduce the number of AppleTalk packets sent to remote network segments, bridging does not reduce the number of broadcast and multicast packets over these networks.

Dynamic startup information

In addition to routing services, the Ascend AppleTalk router provides startup information to AppleTalk stations. Like other routed protocols, AppleTalk station or *node* addresses consist of a unique network-number/node combination. AppleTalk addresses are dynamically assigned when a node starts up. In addition, the router provides an AppleTalk node with the network-cable range to which it is attached and supplies zone-name information.

Configuring AppleTalk routing

To configure AppleTalk routing, you must set system-level parameters in the Ethernet Configuration profile and, if required for caller authentication, in the Answer profile. In addition, you can configure AppleTalk for specific connections.

Ethernet Configuration profile parameters

To set the required parameters in the Ethernet Configuration profile, open the Ether Options submenu of the Mod Config menu and:

1 Enable AppleTalk by setting the AppleTalk parameter to Yes. For example:

```
90-B00 Mod Config
  Ether Options...
    >Domain Name=abc.com
     Pri DNS=200.00.200.14
     Sec DNS=0.0.0.0/0
     Pri DNS=0.0.0.0/0
     Sec DNS=0.0.0.0/0
     SNMP Options
     Route Pref...
     TServ options...
     Bridging=No
     IPX Routing=No
     AppleTalk=Yes
     Shared Prof=No
     Telnet PW=
     RIP Policy=Poison Rvrs
     RIP Summary=Yes
     ICMP Redirect-Accept
     DNS...
     Auth
     Accounting...
```

(You cannot perform the remaining steps of this procedure until you have set AppleTalk to Yes.)

2 In the AppleTalk Options submenu of the Ethernet Configuration profile, set the Zone Name parameter to the name of the zone to which the Ascend unit is connected. Enter up to 33 alphanumeric characters. For example, for router X in Figure 1:

```
90-B00 Mod Config
    AppleTalk Options...
      >Zone Name=SALES
       AppleTalk Router=Seed
       Net Start=300
       Net End=309
       Default Zone=SALES
       Zone Name #1=MKTG
       Zone Name #2=ENGINEERING
       Zone Name #3=
       Zone Name #4=
```

3 In the AppleTalk Options submenu of the Mod Config menu, set the AppleTalk Router parmeter to Seed or Non-Seed to specify whether the Ascend unit is a seed or nonseed router. For example:

```
90-B00 Mod Config
    AppleTalk Options...
      >Zone Name=SALES
       AppleTalk Router=Seed
       Net Start=300
```

```
Net End=309
Default Zone=SALES
Zone Name #1=MKTG
Zone Name #2=ENGINEERING
Zone Name #3=
Zone Name #4=
```

A routed AppleTalk network must include at least one seed router. If you specify Non-Seed, the router learns network number and zone information from other routers. You can set up more than one router on a network to be a seed router, but all seed routers must have the same value for both the start and end of the network number range.

The value zero (0) does not cause a conflict. Non-seed routers and other seed routers can have a value of 0 for the network number range. A router with a value of 0 for a network number range does not send this value to other routers, which means it does not seed the other routers in network with this range. The router with the zero value will not acquire a value for that network number range.

If you specify Non-Seed or Off, skip the remaining steps of this procedure.)

4 If the Ascend unit is a seed router, set the Net Start and Net End parameters to specify the range for the network to which the unit is attached. Valid values are from 1 to 65199. (For example, the menu shown in step 3 specifies a range of 300–309.)

(Remember that all seed routers on the same network must have the same network range.)

5 Specify the default-zone name for the nodes on the seed router's internet. Enter up to 33 alphanumeric characters in the Default Zone field. (For example, the menu shown in step 3 specifies SALES as the default zone.)

All AppleTalk nodes on the seeded network use the default zone until a user explicitely selects a different zone name.

6 Specify the names of the zones that the Ascend unit can seed. Enter up to 33 alphanumeric characters in each of one or more of the Zone Name fields. (For example, the menu shown in step 3 specifies MKTG in the Zone Name #1 field and SALES, MKTG in Zone Name #2.)

A Pipeline can seed up to five zones. A MAX can seed up to 32.

Answer profile parameter

If authentication uses names and passwords, enable AppleTalk routing in the Answer profile by selecting Route AppleTalk=Yes. For example:

```
90-700 Answer
  PPP Options...
   >Route IP=No
    Route IPX=No
    Route AppleTalk=Yes
    Bridge=Yes
    Recv Auth=None
    MRU=1524
```

(You cannot set the Route AppleTalk parameter if AppleTalk is set to No in the Ethernet Configuration profile or if AppleTalk Router is set to Off in that profile's AppleTalk Options submenu.)

Configuring AppleTalk for a specific connection

To enable AppleTalk routing for a particular connection, open the Connection profile and:

1 Select Route AppleTalk=Yes. For example:

```
90-101 Macintosh 1
   >Station=Macintosh 1
     Active=Yes
     Encaps=PPP
     Dial #=1-100-111-1111
     Route IP=No
     Route IPX=No
     Route AppleTalk=Yes
     Bridge=No
     Dial brdcast=N/A
     Encaps options...
     IP options...
     IPX options
     AppleTalk options...
```

(You cannot set the Route AppleTalk parameter if AppleTalk is set to No in the Ethernet Configuration profile or Route AppleTalk is set to No in the Answer profile.)

2 Set the Encaps option to PPP, MPP, or MP. (For example, the Connection profile shown in step 1 specifies PPP for the Encaps option.)

3 In the Dial # field, enter the number to call when this router encounters data destined for the remote Ascend AppleTalk router that is in the zone and network specified in the AppleTalk Options submenu.

4 Select AppleTalk Options to display the AppleTalk Options submenu

5 In the submenu's Zone Name field, enter up to 33 alphanumeric characters to specify the zone name for the AppleTalk router at the other end of the connection. For example:

```
90-101 Macintosh 1
  >AppleTalk options...
    Zone Name=ENGINEERING
    Net Start=2001
    Net End=2010
```

This zone name will appear in the AppleTalk Zones window of the Chooser. If the WAN segment for the zone is not already connected when packets for the zone are received (for example, when a user selects this zone in the Chooser, and then selects AppleShare), the Ascend unit places a call to the number in the Dial # field of the Connection Profile.

6 Enter the network range in the Net Start and Net End fields.

This range defines the networks available for packets that are to be routed to this static route. Valid entries for these fields are in the range from 1 to 65199. If there are other AppleTalk routers on the network, it is necessary to configure the network ranges to coincide with the other routers on the LAN.

Additional information about AppleTalk

This feature note provides only a very brief description of AppleTalk networking. For more complete information, see the following:

Apple Computer. *Inside Macintosh: Networking.*

Chappell, Laura A., and Roger L. Spicer. *Novell's Guide to Multiprotocol Internetworking*.

Sidhu, Andrews, and Alan B. Oppenheimer. *Inside AppleTalk, Second Edition.*

Cougias, Dell, and Heiberger. *Designing AppleTalk Network Architectures.*

Defender authentication added for AppleTalk Remote Access Protocol (ARAP)

A parameter, ARA, has been added to the terminal server to support both AppleTalk Remote Access Protocol (ARAP) and Defender authentication through terminal server. Previously, MAX products supported Defender authentication through the terminal server, but supported ARA through the ARA parameter at Ethernet > Answer > Encaps.

The newer parameter, also called ARA, has been added to the following location:

Ethernet > Mod Config > TServ

Before you use Defender for ARAP

Before you use Defender authentication through the terminal server, you must:

- Have ARA 2.0 or 2.1.

  Note: Due to a Defender limitation, ARA 3.0 and above does not support this new feature.
- Use the reserved name Defender with a blank password in your ARA client software.
- Have Defender security software.
- Have the following scripts:
  - Direct-to-Defender authentication, ASCDIR.SCR
  - RADIUS-to-Defender authentication, ASCAC.SCR.

  These scripts are available on the World Wide Web at http://www.ascend.com or you can contact Ascend Customer Service.
- If you will be using RADIUS to Defender authentication, you must add Authentication-Type = DEFENDER to the RADIUS users file.

ARA

Description: Specifies whether the MAX supports ARAP and Defender authentication through terminal server.

Usage: Specify Yes or No. No is the default.

- Yes enables the MAX terminal server to support ARAP and Defender authentication, provided they meet all other connection criteria.

- No means the MAX will support neither ARAP nor Defender authentication through terminal server. You must change this setting regardless of whether you want to use direct-to-Defender or RADIUS-to-Defender authentication.

Example: ARA=Yes

Dependencies: If you have set AppleTalk or terminal server (TServ) No, ARA becomes not applicable (N/A).

Location: Ethernet > Mod Config > TServ

See Also: AppleTalk, Encaps, ARA Parameter, RADIUS Configuration Guide

Dial-in PPP support for AppleTalk

You can configure an Ascend unit so that individual users can dial into an AppleTalk network using a PPP dialer, such as AppleTalk Remote Access 3.0 and Pacer PPP. The MAX does not need to be set up as an AppleTalk router to support dial-in PPP to AppleTalk.

Overview

The following changes have been made to the user interface to support this feature:

- A new parameter, Peer, has been added to the Connection profile
- A new menu, AppleTalk Options, has been added to the Answer profile.

System requirements

This feature is supported only in MAX system loads that support AppleTalk routing. These loads are indicated by an "a" in the filename (for example, tba.m4, ba.m2).

Note: Many of the MAX updates with AppleTalk routing are "fat loads." For more details refer to the release note entitled "Larger executable load images ("fat loads") enabled."

Configuring dial-in PPP for AppleTalk

You can set up a MAX to allow an AppleTalk client to dial in using PPP in two ways:

- using a Connection profile
- using a Name/Password profile

Configuring an AppleTalk PPP connection using a Connection profile

1. Open the Ethernet > Mod Config menu.
2. Set Appletalk=Yes.
3. Open the appropriate Connection profile.
4. Set Route Appletalk=Yes.
5. Open the AppleTalk options menu.

```
90-103 apple
  AppleTalk options...
```

```
Peer=Dialin
Zone Name=N/A
Net Start=N/A
Net End=N/A
```

6 Set the Peer parameter to indicate whether the connection for this profile is a single user PPP connection or a router

Peer=Dialin indicates that the profile is for a single user PPP connection. All other fields in the AppleTalk options menu are N/A. If you select Peer=Dialin, you have completed the configuration; close the AppleTalk Options menu and save your changes.

Peer=Router indicates that the profile is for a connection with a router (such as an Ascend Pipeline unit). If you select Peer=Router, you will need to configure the other fields in the AppleTalk options menu by continuing with through step 11

Note: Peer=Router works the same way that AppleTalk routing worked before this feature. The following steps are given here for convenience, and duplicate the existing documentation for AppleTalk routing.

7 Configure the AppleTalk zone name for the Ascend unit in the AppleTalk options submenu of the Ethernet Configuration Profile.

If there are other AppleTalk routers on the network, you must configure the zone names and network ranges to coincide with the other routers on the LAN.

The default for the Zone Name field is blank. Enter up to 33 alphanumeric characters to identify the zone name for the unit you are configuring.

Note: These fields will all display N/A if you have not enabled AppleTalk in the Ethernet Mod Config menu.

8 Specify whether the Ascend unit will be a seed or non-seed router. The default value for AppleTalk Router is Off.

- You assign the network range and zone name configuration for a seed router. There must be at least one seed router on a routed AppleTalk network. Select AppleTalk Router=Seed for this option.
- A non-seed router learns network number and zone information from other routers. Select AppleTalk Router=Non-Seed for this option.

  If you choose Non Seed or Off, then Net Start, Net End, Default Zone, and Zone Name #*x* are N/A.

If you are configuring a non-seed router and are using Names/Passwords, go to Configuring an AppleTalk PPP connection using a Name/Password profile.

9 If you are configuring the Ascend unit as a seed router, specify the network range for the network to which the Ascend unit is attached.

Net Start and Net End define the network range for nodes attached to this network. Valid entries for these fields are in the range from 1 to 65199. If there are other AppleTalk routers on the network, you must configure the network ranges to coincide with the other routers.

10 Specify the default zone name for nodes on the Ascend unit's internet.

Enter up to 33 alphanumeric characters for the default zone name. The default for this field is blank.

The default zone is the one used by a node in the network for which you are configuring the Connection Profile until another zone name is explicitly selected by the node.

11 Specify the zone names that the platform can seed.

The Pipeline can seed up to 5 zones, and the MAX can seed up to 32. Enter up to 33 alphanumeric characters in zone name fields.

Configuring an AppleTalk PPP connection using a Name/Password profile

1. Open the Ethernet > Mod Config menu.
2. Set Appletalk=Yes.
3. Open the PPP Options menu of the Answer profile.
4. Set Route Appletalk=Yes.
5. Open the Appletalk options submenu of the PPP options menu.

```
90-103 apple
  AppleTalk options...
   Peer=Dialin
```

6. Set the Peer parameter to indicate whether the connection for this profile is a single user PPP connection or a router

   Peer=Dialin indicates that the profile is for a single user PPP connection. All other fields in the AppleTalk options menu are N/A. If you select Peer=Dialin, you have completed the configuration; close the AppleTalk Options menu and save your changes.

   Peer=Router indicates that the profile is for a connection with a router (such as an Ascend Pipeline unit). If you select Peer=Router, you will need to configure the other fields in the AppleTalk options menu by continuing with Step 7 through Step 11 in Configuring an AppleTalk PPP connection using a Name/Password profile.

   Note: Peer=Router works the same way that AppleTalk routing worked before this feature. The Step 7 through Step 11 are given here for convenience, and duplicate the existing documentation for AppleTalk routing.

SecurID authentication for AppleTalk Remote Access (ARA) users

This release enables ARA callers to use AppSecurID authentication by contacting an ACE server through a Connection profile, Password profile, or RADIUS user profile. Previously, ARA callers could use only direct name and password authentication without the use of an external authentication server.

How an ARA caller uses SecurID authentication

An ARA caller can use SecurID authentication in any of the following ways:

- Using a Connection profile
- Using a Password profile
- Using a RADIUS user profile

  If the user has a RADIUS user profile, he or she must have the username "SecurID".

For information on setting up a profile to contact an external authentication server, including an ACE server, see the *MAX Security Supplement*.

Once the user makes the initial connection, SecurID authentication begins with a pop-up screen on the Macintosh. At this point, the user must enter the "User ID" and "Passcode". If the user enters incorrect values, he or she gets two more tries to authenticate before the connection fails.

If the user is required to enter a new PIN, a pop-up screen prompts for this information. The user has three chances to enter the correct PIN. Once the new PIN is accepted, a pop-up screen instructs the Macintosh user to wait for the token code to change and then to log in with the new PIN and token code.

The SecurID client module must be version 1.3 or later.

Modem features

Megahertz CC1336 and XJ1336 modems available with MAX 200Plus

You can now useMegahertz CC1336 and Megahertz XJ1336 modems in a MAX 200Plus.

Overview

You can install up to eight supported Megahertz CC1336 and Megahertz XJ1336 modems in a MAX 200Plus. These modems can be used for

- incoming connections to the MAX 200Plus
- outgoing connections from the MAX 200Plus (using MAXDial)

The MAX 200Plus now supports Megahertz CC1336 and XJ1336 modem cards with firmware version 2.4.

How the MAX 200Plus supports the modem

The MAX 200Plus supports Megahertz CC1336 and Megahertz XJ1336 modem by

- initializing the modem when necessary
- identifying the modem

Initialization string

When the MAX 200Plus initializes the modem, it sends it the following initialization string:

```
AT&F1E0S27=16S0=1
```

Modem identification in the Telnet interface

After you install the new modem, the name of the modem manufacturer is shown as the value of the PC CARD Modem > Mod Config > Name parameter for the card slot containing the

modem. The name of the modem model is shown as the value of the PC CARD Modem > Mod Config > Product parameter.

Modem identification in the MAX 200Plus Console

After you install the new modem, the Ports tab of the MAX 200Plus Console program shows the name of the modem manufacturer.

USR Courier V.Everything modem available with MAX 200Plus

You can now useUSR Courier V.Everything modems in a MAX 200Plus.

Overview

You can install up to eight supported PC Card (PCMCIA) modems in a MAX 200Plus. These modems can be used for

- incoming connections to the MAX 200Plus
- outgoing connections from the MAX 200Plus (using MAXDial)

The MAX 200Plus now supports the USR Courier V.Everything modem card with firmware version 7.1.6.

How the MAX 200Plus supports the modem

The MAX 200Plus supports theUSR Courier V.Everything modem by

- initializing the modem when necessary
- identifying the modem

Initialization string

When the MAX 200Plus initializes the modem, it sends it the following initialization string:

```
AT&F1E0S27=16S0=1
```

Modem identification in the Telnet interface

After you install the new modem, the name of the modem manufacturer is shown as the value of the PC CARD Modem > Mod Config > Name parameter for the card slot containing the modem. The name of the modem model is shown as the value of the PC CARD Modem > Mod Config > Product parameter.

Modem identification in the MAX 200Plus Console

After you install the new modem, the Ports tab of the MAX 200Plus Console program shows the name of the modem manufacturer.

Support for Xircom 33.6 PC Card modems

You can now use Xircom 33.6 PC Card (PCMCIA) modems in a MAX 200Plus.

Overview

You can install up to eight supported PC Card (PCMCIA) modems in a MAX 200Plus. These modems can be used for

- incoming connections to the MAX 200Plus
- outgoing connections from the MAX 200Plus (using MAXDial)

The MAX 200Plus now supports the Xircom 33.6 PC Card modem (model number CM-33(AM)) with firmware version 4.15.

How the MAX 200Plus supports the modem

The MAX 200Plus supports the Xircom 33.6 PC Card modem by

- initializing the modem when necessary
- identifying the modem

Initialization string

When the MAX 200Plus initializes the modem, it sends it the following initialization string:

```
AT&FW2X4&C1&D3&Q5%C1S36=3S7=60S0=1
```

Manual initialization

Version 5.0Ai3 and later of the MAX 200 software automatically initializes the modem when necessary. If you have a MAX 200Plus with software earlier than version 5.0Ai3, you can initialize the modem by using a terminal or other communication program to send it the initialization string. No other initialization is necessary for this modem.

Modem identification in the Telnet interface

After you install the new modem, the name of the modem manufacturer is shown as the value of the PC CARD Modem > Mod Config > Name parameter for the card slot containing the modem. The name of the modem model is shown as the value of the PC CARD Modem > Mod Config > Product parameter.

Modem identification in the MAX 200Plus Console

After you install the new modem, the Ports tab of the MAX 200Plus Console program shows the name of the modem manufacturer.

Fax mode limitation

Version 4.15 of the Xircom 33.6 PC Card firmware does not support Class 2 fax operation.

Support for Hayes Optima 33.6 PC Card (PCMCIA) modems

You can now use Hayes Optima 33.6 PC Card (PCMCIA) modems in a MAX 200Plus.

Overview

You can install up to eight supported PC Card (PCMCIA) modems in a MAX 200Plus. These modems can be used for

- incoming connections to the MAX 200Plus
- outgoing connections from the MAX 200Plus (using MAXDial)

The MAX 200Plus now supports the Hayes Optima 33.6 PC Card modem (model number 5346US) with firmware version 4.4.

How the MAX 200Plus supports the modem

The MAX 200Plus supports the Optima 33.6 PC Card modem by

- initializing the modem when necessary
- identifying the modem

Initialization string

When the MAX 200Plus initializes the modem, it sends it the following initialization string:

```
AT&FE0W2&C1&K3&Q5S7=60S46=2S48=7S36=3S0=1
```

Manual initialization

Version 5.0Ai3 and later of the MAX 200 software automatically initializes the modem when necessary. If you have a MAX 200Plus with software earlier than version 5.0Ai3, you can initialize the modem by using a terminal or other communication program to send it the initialization string. No other initialization is necessary for this modem.

Modem identification in the Telnet interface

After you install the new modem, the name of the modem manufacturer is shown as the value of the PC CARD Modem > Mod Config > Name parameter for the card slot containing the modem. The name of the modem model is shown as the value of the PC CARD Modem > Mod Config > Product parameter.

Modem identification in the MAX 200Plus Console

After you install the new modem, the Ports tab of the MAX 200Plus Console program shows the name of the modem manufacturer.

New modems supported on the MAX 200Plus

The MAX 200Plus now includes support for the following modems: U.S. Robotics SP1336, Megahertz CC1556, Megahertz XJ1556, E-Tech C336MX Bullet, Aiwa PV-JF3360, Omron ME3314C, and TDK DF3314E.

Overview

You can install up to eight supported PC Card (PCMCIA) modems in a MAX 200Plus. These modems can be used for

- incoming connections to the MAX 200Plus
- outgoing connections from the MAX 200Plus (using MAXDial)

How the MAX 200Plus supports these modems

The MAX 200Plus now includes support for the following modems:

- U.S. Robotics
  Model: Sportster
  Model Number: SP 1336 (non-interfering)
  Firmware version: 2.4
- Megahertz
  Model: CC1556 (non-interfering)
  Model: XJ1556 (interfering)
  Model Number: NA
  Firmware version: 10.1.24
- E-Tech
  Model: Bullet (non-interfering)
  Model Number: C336MX
  Firmware version: 9.12
  (certified for use in Taiwan and North America)
- Aiwa
  Model: N/A
  Model Number: PV-JF3360
  Firmware version: N/A
- Omron
  Model: N/A
  Model Number: ME3314C
  Firmware version: J1.510
- TDK
  Model: N/A
  Model Number: DF3314E
  Firmware version: 1.61a-J
  (non-interfering, Japan only)

The MAX 200Plus supports these modems by:

- initializing the modem when necessary
- identifying the modem

Initialization strings

When the MAX 200Plus initializes the modem, it sends it the following initialization strings:

- For the U.S. Robotics Sportster (SP1336)
  AT&F1E0S15=8S0=1
- For the Megahertz CC1556 and the Megahertz XJ1556
  AT&F1E0S15=8S0=1
- For the E-Tech C336MX Bullet
  AT&F\V1W2S0=1
- For the Aiwa PV-JF3360
  AT&F%C2\V2E0S0=1
- For the Omron ME3314C
  `AT&F%C2W2E0S0=1`
- For the TDK DF3314E
  AT&F%C2W2E0S0=1

Initializing the modem manually

The MAX 200Plus software automatically initializes the modem if it is on the approved modem list. If you wish to use a newly approved modem, you may either install this software version, or enter the initialization string manually. Instructions for entering initialization strings appear on the MAX 200Plus Approved Modem list (refer to the Ascend Home Page.)

Identifying the modem in the Telnet interface

After you install the new modem, the name of the modem manufacturer appears as the value of the PC CARD Modem > Mod Config > Name parameter for the card slot containing the modem. The name of the modem model appears as the value of the PC CARD Modem > Mod Config > Product parameter.

Identifying the modem on the MAX 200Plus Console

After you install the new modem, the Ports tab of the MAX 200Plus Console program shows the name of the modem manufacturer.

Indicating a supported modem on the MAX 200Plus

On the front panel of the MAX 200Plus, the green ON LED displays continuously to indicate support for the inserted modem.

Support for Practical Peripherals 33.6 V.34+Fax PC

Card modems

You can now use a Practical Peripherals 33.6 V.34+Fax PC Card (PCMCIA) modem in a MAX 200Plus.

Overview

You can install up to eight supported PC Card (PCMCIA) modems in a MAX 200Plus. These modems can be used for

- incoming connections to the MAX 200Plus
- outgoing connections from the MAX 200Plus (using MAXDial)

The MAX 200Plus now supports the Practical Peripherals 33.6 V.34+Fax PC Card modem (model number 5352US) with firmware version 4.4.

How the MAX 200Plus supports the modem

The MAX 200Plus supports the Practical Peripherals 33.6 V.34+Fax PC Card modem by

- initializing the modem when necessary
- identifying the modem

Initialization string

When the MAX 200Plus initializes the modem, it sends it the following initialization string:

```
AT&FE0W2&C1&Q5&K3S7=60S36=3S46=2S48=7S95=1S0=1
```

Manual initialization

Version 5.0Ai3 and later of the MAX 200 software automatically initializes the modem when necessary. If you have a MAX 200Plus with software earlier than version 5.0Ai3, you can initialize the modem by using a terminal or other communication program to send it the initialization string. No other initialization is necessary for this modem.

Modem identification in the Telnet interface

After you install the new modem, the name of the modem manufacturer is shown as the value of the PC CARD Modem > Mod Config > Name parameter for the card slot containing the modem. The name of the modem model is shown as the value of the PC CARD Modem > Mod Config > Product parameter.

Modem identification in the MAX 200Plus Console

After you install the new modem, the Ports tab of the MAX 200Plus Console program shows the name of the modem manufacturer.

Support for Practical Peripherals 33.6 PC Card modem

with EZ-Port

You can now use a Practical Peripherals 33.6 PC Card (PCMCIA) modem with EZ-Port in a MAX 200Plus.

Overview

You can install up to eight supported PC Card (PCMCIA) modems in a MAX 200Plus. These modems can be used for

- incoming connections to the MAX 200Plus
- outgoing connections from the MAX 200Plus (using MAXDial)

The MAX 200Plus now supports the Practical Peripherals 33.6 PC Card modem with EZ-Port (model number 5353US) with firmware version 4.4.

How the MAX 200Plus supports the modem

The MAX 200Plus supports the Practical Peripherals 33.6 PC Card modem with EZ-Port by

- initializing the modem when necessary
- identifying the modem

Initialization string

When the MAX 200Plus initializes the modem, it sends it the following initialization string:

```
AT&FE0W2&C1&Q5&K3S7=60S36=3S46=2S48=7S95=1S0=1
```

Manual initialization

Version 5.0Ai3 and later of the MAX 200 software automatically initializes the modem when necessary. If you have a MAX 200Plus with software earlier than version 5.0Ai3, you can initialize the modem by using a terminal or other communication program to send it the initialization string. No other initialization is necessary for this modem.

Modem identification in the Telnet interface

After you install the new modem, the name of the modem manufacturer is shown as the value of the PC CARD Modem > Mod Config > Name parameter for the card slot containing the modem. The name of the modem model is shown as the value of the PC CARD Modem > Mod Config > Product parameter.

Modem identification in the MAX 200Plus Console

After you install the new modem, the Ports tab of the MAX 200Plus Console program shows the name of the modem manufacturer.

Support for the Hayes Optima 33.6 PC Card modem

with EZjack

You can now use a Hayes Optima 33.6 PC Card modem with EZjack in a MAX 200Plus.

Overview

You can install up to eight supported PC Card (PCMCIA) modems in a MAX 200Plus. These modems can be used for

- incoming connections to the MAX 200Plus
- outgoing connections from the MAX 200Plus (using MAXDial)

The MAX 200Plus now supports the Hayes Optima 33.6 PC Card modem with EZjack (model number 5347US) with firmware version 4.4.

How the MAX 200Plus supports the modem

The MAX 200Plus supports the Optima 33.6 PC Card modem with EZjack by

- initializing the modem when necessary
- identifying the modem

Initialization string

When the MAX 200Plus initializes the modem, it sends it the following initialization string:

```
AT&FE0W2&C1&K3&Q5S7=60S46=2S48=7S36=3S0=1
```

Manual initialization

Version 5.0Ai3 and later of the MAX 200 software automatically initializes the modem when necessary. If you have a MAX 200Plus with software earlier than version 5.0Ai3, you can initialize the modem by using a terminal or other communication program to send it the initialization string. No other initialization is necessary for this modem.

Modem identification in the Telnet interface

After you install the new modem, the name of the modem manufacturer is shown as the value of the PC CARD Modem > Mod Config > Name parameter for the card slot containing the modem. The name of the modem model is shown as the value of the PC CARD Modem > Mod Config > Product parameter.

Modem identification in the MAX 200Plus Console

After you install the new modem, the Ports tab of the MAX 200Plus Console program shows the name of the modem manufacturer.

Support for Hayes Optima 288 V.34+Fax PC Card

modems (Australia)

You can now use Hayes Optima 288 V.34+Fax PC Card (PCMCIA) modems for Australia in a MAX 200Plus.

Overview

You can install up to eight supported PC Card (PCMCIA) modems in a MAX 200Plus. These modems can be used for

- incoming connections to the MAX 200Plus
- outgoing connections from the MAX 200Plus (using MAXDial)

The MAX 200Plus now supports the Hayes Optima 288 V.34+Fax PC Card modem for Australia (model number 08-02546) with firmware version 7.173.

How the MAX 200Plus supports the modem

The MAX 200Plus supports the Optima V.34+Fax PC Card modem by

- initializing the modem when necessary
- identifying the modem

Initialization string

When the MAX 200Plus initializes the modem, it sends it the following initialization string:

```
AT&FE0W2&C1&K3&Q5S7=60S46=2S48=7S36=3S0=1
```

Other initialization

During initialization, the MAX 200 must also set the Configuration Option register of the modem. Because of this, you cannot initialize the modem by sending it only an initialization string. In addition, you must use version 5.0Ai3 or later of the MAX 200Plus software to use the modem; earlier versions of the software cannot set this register.

Modem identification in the Telnet interface

After you install the new modem, the name of the modem manufacturer is shown as the value of the PC CARD Modem > Mod Config > Name parameter for the card slot containing the modem. The name of the modem model is shown as the value of the PC CARD Modem > Mod Config > Product parameter.

Modem identification in the MAX 200Plus Console

After you install the new modem, the Ports tab of the MAX 200Plus Console program shows the name of the modem manufacturer.

Fax mode limitation

When the Optima V.34+Fax PC Card modem is in fax mode, you cannot include spaces in the Calling Station ID for the modem.

Flashing CD lights indicate problems with modems

Flashing red CD (Carrier Detect) lights on the front panel of the MAX 200Plus indicate a modem problem.

A modem that fails to reset after an error has occurred, displays flashing red CD lights. These lights generally indicate a problem with the modem software but can occasionally indicate a problem with modem or slot hardware.