Ascend NavisRadius User’s Guide

Ascend Communications

Part Number 7820-0380-001 November 2, 1998

ASCEND END USER AGREEMENT

License

The term “Software” includes all Ascend and third party (“Supplier”) software provided to you with this Ascend product, and includes any accompanying documentation (the “Documentation”). The term “Software” also includes any updates of the Software provided to you by Ascend at its option. Subject to the terms of this Agreement, Ascend grants to you, and you accept, a personal, non-exclusive, and nontransferable (except as set forth below) license to use the object code version of the Software on a single computer. The Software is “in use” on a computer when it is loaded into the temporary memory (i.e. RAM) or installed into the permanent memory (e.g. hard drive, CD-ROM or other storage device) of that computer, except that a copy installed on a network server for the sole purpose of distribution to other computers is not “in use.” If you permanently install the Software on the hard disk or other storage device of a computer (other than a network server) and you use that computer more than 80% of the time it is in use, then you may also use the Software on a portable or home computer. You may make a reasonable number of copies of the Software and Documentation for backup or archival purposes only, so long as Ascend's and its licensors’ copyright notices are reproduced on such copies.

Limitations on Use

You may not copy, rent, lease, sell, sublicense, assign, loan, time-share or otherwise transfer or distribute copies of the Software or Documentation to others, except as set forth in this agreement. You may physically transfer the Software from one computer to another provided that you do not retain any copies of the Software, including any copies stored on a computer. You may permanently transfer this license to another user, but only if you transfer or destroy all copies of the Software and Documentation, and the recipient agrees in writing to be bound by all of the terms of this agreement.

You agree that you will not decompile, disassemble, or otherwise reverse engineer the Software, and you will use your best efforts to prevent your employees and contractors from doing so, except to the extent that such restriction is expressly prohibited by applicable law. You may not modify, adapt, create a derivative work, merge, or translate the Software or the Documentation without the prior written consent of Ascend.

Specific Suppliers may be identified in the Documentation. You agree to any additional terms and conditions specific to particular Suppliers or Products, as described in the Documentation, which are incorporated herein by reference.

Intellectual Property Rights

You acknowledge that Ascend or its Suppliers retain exclusive ownership of all copyrights, trademarks, patents and/or other intellectual property rights in the Software and the Documentation. You are not granted any rights in the Software or Documentation other than the license rights expressly set forth above.

Term and Termination

The term of this license is for the duration of any copyright in the Software. This license automatically terminates if you fail to comply with any of the terms and conditions of this agreement. You agree that, upon such termination, you will either destroy (or permanently erase) all copies of the Software and Documentation, or return the original Software and Documentation to Ascend. You may terminate this license at any time by destroying the Software and Documentation and any permitted copies.

Limited Warranty and Limited Remedy

Ascend warrants to the original end user purchaser only that the Software as delivered at the time of purchase will substantially conform to the Documentation, and that the original diskettes and Documentation are free from defects in material and workmanship under normal use, for a period of ninety (90) days from the original end user's purchase thereof (the “Limited Warranty Period”), provided the Software is used with compatible computer hardware and operating systems. This limited warranty is void if failure of the Software has resulted from accident, abuse, or misapplication. Ascend's entire liability, and your sole and exclusive remedy shall be, at Ascend's option, either to (a) correct or help you work around or avoid a reproducible Error, (c) replace defective diskettes or Documentation or (b) authorize a refund, so long as the Software and Documentation are returned with a copy of your receipt within ninety (90) days of your date of purchase together with a brief written statement describing the alleged Error. An “Error” is a defect in the Software that causes it not to perform substantially in accordance with the limited warranty set forth above. Any replacement Software or Documentation will be warranted for the remainder of the original warranty period only.

No Liability of Suppliers

You acknowledge that your rights under this Agreement, in the nature of warranty or otherwise, are solely against Ascend. NO SUPPLIER MAKES ANY WARRANTY, ASSUMES ANY LIABILITY, OR UNDERTAKES TO FURNISH TO YOU ANY SUPPORT OR INFORMATION CONCERNING PRODUCTS OR ANY PORTION OF PRODUCTS. You hereby release all Suppliers from any claims, damages or losses arising from the use of Products, regardless of the form of action.

Disclaimer of Warranties

EXCEPT AS SPECIFICALLY PROVIDED ABOVE, THE SOFTWARE AND THE DOCUMENTATION IS PROVIDED “AS IS,” WITHOUT WARRANTY OF ANY KIND. ALL OTHER WARRANTIES ARE DISCLAIMED, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT OR THAT THE SOFTWARE'S FUNCTIONS WILL MEET YOUR REQUIREMENTS OR THAT ITS OPERATION WILL BE UNINTERRUPTED OR ERROR FREE. EXCEPT AS SET FORTH IN THIS AGREEMENT, THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE AND THE DOCUMENTATION IS WITH YOU. IF THEY PROVE DEFECTIVE AFTER THEIR PURCHASE, YOU, AND NOT ASCEND OR ITS SUPPLIERS, ASSUME THE ENTIRE COST OF SERVICE OR REPAIR. If a disclaimer of implied warranties is not permitted by law, the duration of any such implied warranty is limited to ninety (90) days from the date of purchase by the original end user purchaser. Some jurisdictions do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, so such limitations or exclusions may not apply to you. This limited warranty gives you specific legal rights, and you may also have other rights which vary from jurisdiction to jurisdiction.

Liability Exclusions and Limitations

IN NO EVENT SHALL ASCEND OR ANY SUPPLIER BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES OF ANY KIND (INCLUDING LOST PROFITS, LOSS OF USE OR INTERRUPTION OF BUSINESS), OR FOR LEGAL FEES, ARISING OUT OF THE USE OF THE SOFTWARE OR THE DOCUMENTATION, REGARDLESS OF THE FORM OF ACTION, WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT PRODUCT LIABILITY OR OTHERWISE, EVEN IF ASCEND HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL ASCEND'S AGGREGATE LIABILITY FOR ANY CLAIM EXCEED THE LICENSE FEE PAID BY YOU. This limitation shall apply notwithstanding any failure or inability to provide the limited remedies set forth above. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation(s) or exclusion(s) may not apply to you.

Proprietary Rights-Contracts with Certain U.S. Government Agencies

If the Software is acquired under the terms of a Department of Defense or civilian agency contract, the Software is “commercial item” as that term is defined at 48 C.F.R. 2.101 (Oct. 1995), consisting of “commercial computer software” and “commercial computer software documentation” as such terms are used in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995) of the DoD FAR Supplement and its successors. All U.S. Government end users acquire the Software and the Documentation with only those rights set forth in this agreement.

Export Restrictions

You acknowledge that the laws and regulations of the United States restrict the export and re-export of certain commodities and technical data of United States origin, including the Software and the Documentation, in any medium. You agree that you will not knowingly, without prior authorization if required, export or re-export the Software or the Documentation in any medium without the appropriate United States and foreign government licenses.

Severability

You acknowledge and agree that each provision of this agreement that provides for a disclaimer of warranties or an exclusion or limitation of damages represents an express allocation of risk, and is part of the consideration of this agreement. Invalidity of any provision of this Agreement shall not affect the validity of the remaining provisions of this Agreement.

General

This Agreement is the entire agreement between you and Ascend relative to the Software and Documentation, and supersedes all prior written statements, proposals or agreements relative to its subject matter. It may be modified only by a writing executed by an authorized representative of Ascend. No Ascend dealer or sales representative is authorized to make any modifications, extensions or additions to this agreement. This Agreement is governed by the laws of the State of California as applied to transactions taking place wholly within California between California residents, without application of its conflicts of law principles. The United Nations Convention on Contracts for the International Sale of Goods is specifically excluded from application to this Agreement.

Questions

If you have any questions, write or call Ascend Communications, Inc., One Ascend Plaza, 1701 Harbor Bay Parkway, Alameda, CA 94502.

Ascend Customer Service

When you contact Ascend Customer Service, make sure you have this information:

- The product name and model
- The software and hardware options
- The software version
- The type of computer you are using
- A description of the problem

How to contact Ascend Customer Service

Ways to contact Ascend Customer Service	**Telephone number or address**
Telephone in the United States	800-ASCEND-4 800-272-3634
Telephone outside the United States	510-769-8027
E-mail	support@ascend.com
Facsimile (FAX)	510-814-2300

You can also call Ascend Communications main office at 510-769-6001 or write to Ascend at:

Ascend Communications
1275 Harbor Bay Parkway
Alameda, CA 94502

Need information on new features and products?

We are committed to constantly improving our products. You can find out about new features and product improvement by visiting 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

Chapter 5 Understanding NavisRadius........ 5-1

About This Guide

How to use this guide

This guide describes Ascend NavisRadius, a software package that provides authentication, authorization, and accounting services for users who request network connections. Ascend NavisRadius includes a RADIUS server.

This guide contains instructions for configuring files that contain data Ascend NavisRadius must parse to perform authentication and authorization. Ascend NavisRadius is distributed with example configuration files you can copy. Edit the copied file’s entries to reflect your personal installation.

What you should know

This guide is intended for the person who will configure and maintain an Ascend NavisRadius server. You must have a basic understanding of security, authentication methods, and networking concepts.

Documentation conventions

Table 1.NavisRadius User Guide conventions.

Convention	**Meaning**
Monospace text	Represents text that appears on your computer's screen, or that could appear on your computer's screen.
`Boldface monospace text`	Represents characters that you enter exactly as shown (unless the characters are also in *`italics`*—see *Italics*, below). If you could enter the characters, but are not specifically instructed to, they do not appear in boldface.
Italics	Represent variable information. Do not enter the words themselves in the command. Enter the information they represent. In ordinary text, italics are used for titles of publications, for some terms that would otherwise be in quotation marks, and to show emphasis.
[ ]	Square brackets indicate an optional argument you might add to a command. To include such an argument, type only the information inside the brackets. Do not type the brackets unless they appear in bold type.
|	Separates command choices that are mutually exclusive.
>	Points to the next level in the path to a parameter. The parameter that follows the angle bracket is one of the options that appears when you select the parameter that precedes the angle bracket.
Key1-Key2	Represents a combination keystroke. To enter a combination keystroke, press the first key and hold it down while you press one or more other keys. Release all the keys at the same time. (For example, Ctrl-H means hold down the Control key and press the H key.)
Press Enter	Means press the Enter, or Return, key or its equivalent on your computer.
Note:	Introduces important additional information.

Convention	Meaning
⚠ **Caution:**	Warns that a failure to follow the recommended procedure could result in loss of data or damage to equipment.
⚡ **Warning:**	Warns that a failure to take appropriate safety precautions could result in physical injury.

Related publications

If your network includes Ascend routers that support Ascend NavisRadius, the supplemental security sections in your user guides contain useful information about RADIUS and other topics in the Ascend Navis Radius User’s Guide.

Introducing Ascend NavisRadius

1

This chapter lists Ascend NavisRadius's features and provides an introduction to the Ascend NavisRadius authentication, authorization and accounting functions .

What is Ascend NavisRadius

The Ascend NavisRadius server is part of the Ascend Navis family of software products developed to manage your network. Ascend NavisRadius provides authentication, authorization, and accounting services. It is based on the RADIUS protocol.

NavisRadius provides a central location to store information, or attributes, that defines individual users and the local and remote systems and services you allow them to access. The attributes and their values are maintained in user profiles. You can store user profiles on the NavisRadius servers, on other servers for which NavisRadius can be a proxy, or in the NavisRadius Database Management System (DBMS) tables. NavisRadius's user profile information enables you to:

- Authenticate users
- Configure incoming WAN connections
- Configure dialout connections
- Establish routes
- Install filters
- Record connection-accounting data

Ascend products supported by Ascend NavisRadius

NavisRadius is extensible. Its `dictionary` file contains all standard RADIUS attributes and many others developed by Ascend for these units:

- MAX TNT
- MAX 6000
- MAX 4000, 4002/4004, 4048/4060
- MAX 2000, 2012/2014
- MAX 1800
- MAX 200Plus
- Pipeline 400
- Pipeline 220

Ascend NavisRadius features

NavisRadius is a packaged executable. You do not need to compile source code before installing the program. Some of the features it supports are:

- Licensed support for security tokens
  - Security Dynamics ACE
  - Axent Technology Defender
  - Bellcore S/Key
  - Token card caching
- Password aging
- Proxy-RADIUS authentication and accounting
- Multiple realms for authenticating users by domain
- Vendor-specific user profile attributes
- User Authentication via other methods
  - TACACS and TACACS+
  - MIT and AFS Kerberos v4 (see Note)
- RADIUS IP Address Daemon (RADIPAD)
- RADIUS user STATE Daemon (RADSTATED)
- Ascend packet filter profiles
- Global IP pools
- iPass Global Roaming
- SQL compliant runtime DBMS

Note: Some features are not available on all operating systems. Please note these restrictions:

- Kerberos authentication is not supported on Windows/NT
- i-Pass Global proxy is not supported on Windows/NT
- NavisRadius is supplied with an integrated runtime DBMS which is Sybase SQL Anywhere version 5 server.

What Ascend NavisRadius does

NavisRadius performs three functions. It authenticates and authorizes remote users and, optionally, stores accounting information about the events that occur during users' connections to the network.

Authentication

NavisRadius's primary function is to confirm that someone requesting a connection to or from the network has permission to make the connection and to use network services. NavisRadius authenticates users by comparing information the users send to Network Access Servers (NAS) with information called check-items stored the user profiles in NavisRadius' `users` files.

NavisRadius also authenticates the NAS machines that send it Access-Request packets on behalf of the users that want to access the network. NavisRadius and the NAS share a secret key that allows the NavisRadius server to identify the NAS. The shared secret is stored as the value of a field in NavisRadius `clients` file entries.

Authorization

NavisRadius `users` files can contain reply-items as well as check-items, such as names, passwords and secret keys. Reply-items define the services users can access and the connections they can make. NavisRadius may include reply-items from user profiles in Access-Response packets it sends to clients that request user-authentication. The reply-items may authorize the NAS to allow the user to connect to a specific network machine, such as a Telnet server.

Authorization information that NavisRadius sends to a NAS might also tell the NAS how users are allowed to make internal and external connections. For example, NavisRadius's authorization might instruct the NAS to allow an incoming connection to a local Telnet server via an unframed protocol, but force another, outbound, user to connect with a remote location via the framed Point-to-Point Protocol.

Accounting

NavisRadius's third function is to serve as a repository of information about the activities that occur over network connections. When it has authenticated a user and authorized the connection and services, NavisRadius might receive an accounting of the connection from the NAS. The accounting can include such things as the start and stop times of the connection and the number of packets and octets that passed over the connection. Information such as this is valuable if you want to evaluate your security policies and network usage.

How Ascend NavisRadius works

Although simplified, the following steps summarize how NavisRadius works:

1 A client, such as a router, contacts the NavisRadius server and sends NavisRadius a message requesting authentication of a user who has requested network access. The outside source that contacts NavisRadius is referred to as a client.

2 NavisRadius searches its `clients` file for an entry that matches client-identification information in the client's message. If NavisRadius finds a match, it extracts user-identification information from the message. Typically, the user-identification in the message is a user name and a password or type of authentication.

3 NavisRadius searches a `users` file for a profile that contains data that matches the user-identification information from the client's request message. If successful, NavisRadius sends the client a user-authentication message.

4 The message verifies the user's identity, specifies the extent of the user's access to the network, and defines how the user's link to the network is to be configured.

Authentication and authorization information

You need to supply NavisRadius with information it can use to authenticate and authorize, including:

- Who its clients and their users are

- What its clients and their users will provide as identifying information
- Where to find matching information to identify clients and their users
- How its clients should configure their users' access and links

You store the information in NavisRadius files named `clients` and `users`. You might also store some of the information you gather in an NavisRadius file named `authfile` if you want to group users in entities called realms.

Gathering needed information

Gathering the needed information and adding it to the appropriate files is less complicated than it might appear. Suppose you receive a call from a city building inspector who requests that you meet and allow him to see some of your company's site plans for a new project. You ask the inspector some questions, while mentally considering some of the issues and consequences of granting his request. Here are your questions and considerations:

- Who are you?
- How will I know you?
- Where are you located?
- What's your telephone number?
- Should I call him back to verify that number?
- Where do you want to go?
- How do you want to get there?
- What do you want to see there?
- How can I limit what he sees?
- How long will you be involved in the project?
- Should I keep track of him while he's there?

The caller answers your questions. His name is Jeff Carr and his telephone number is 212-555-1212. Tomorrow he will drive to your office from the city building at 929 High Street. He will show the guard his employee ID so the guard can bring him to your office. After reviewing the plans in your office, he would like to visit the construction site next door. He will probably return every few weeks for six months to keep abreast of the project's progress.

After hanging up you decide that, for now, you need only show him the model of the project. You think it will be a good idea to have a guard accompany him on his visit to the construction site and report back to you afterwards. You will provide Mr. Carr a pass to the site that is good for 30 days. Later you will call the number he gave you and ask to speak to him to confirm the time of your appointment.

At this point, you have gathered many of the kinds of information NavisRadius uses to authenticate and authorize a user. Table 1-1 illustrates how the building inspector's answers and your own security plan have given you information that resembles the values of attributes you enter in an NavisRadius user profile.

Table 1-1. Identifying Ascend NavisRadius attribute values

Question	**Ascend NavisRadius Attribute**
Who are you?	User-Name
How will I know you?	Password
Where are you located?	Framed-IP-Address
What's your telephone number?	Calling-Station-ID
Should I call you back to verify your identity?	Ascend-Callback
Where do you want to go?	NAS-IP-Address
How do you want to get there?	Framed-Protocol
What do you want to see there?	Service-Type
How can I limit what you see?	Ascend-Data-Filter
How long will you be involved in the project?	Expiration
Should I keep track of you while you're there?	Accounting service

NavisRadius includes `dictionary`, a file that lists all the types of information you can collect about users and their connections. NavisRadius uses the term attribute to describe a type of information. Each attribute has a value, or a list of possible values. You could think of the name “Jeff Carr” as the value of the User-Name attribute displayed in parentheses in the table’s right column.

NavisRadius uses some attributes, such as User-Name, to perform authentication. It uses others, like Framed-Protocol, to perform authorization. NavisRadius attributes and their values are described in the Reference section of the Appendix.

Compatibility with Ascend RADIUS

The Ascend RADIUS server source code is offered as a public (non-commercial) code. NavisRadius is a commercial product. Ascend RADIUS and NavisRadius are compatible, although some of the differences in the programs are noted in the sections that follow. The items listed in “Attribute differences,” are explained in Chapter 6, “Creating Example Client, User and Realm entries.” Appendix A, “Ascend NavisRadius files and commands,” discusses the items listed in “Operational differences.”

Note about the RADIUS builddbm utility

NavisRadius does not support `builddbm`, a utility that creates an index of user profiles in the `users` file. When the `users` file has been indexed by `builddbm` the RADIUS daemon can more quickly search the file for a matching user profile. RADIUS programs that you must compile might permit you to use `builddbm` before you compile the program’s source code.

NavisRadius does not support indexing and the `builddbm` utility because NavisRadius is delivered as a binary application, ready to install and run. You do not have to compile NavisRadius. Moreover, NavisRadius supports multiple `users` files and ODBC compliant DBMS, which provide much more powerful and flexible solutions for authenticating and authorizing large numbers of users than does a single, indexed text file.

Attribute differences

- Changing passwords
  - A new name is assigned to an attribute that defines how a user can change passwords during authentication
  - The Ascend NavisRadius RADIUS daemon requires a new command line option, `-P`
- Specifying a Token Card
  - Ascend NavisRadius is not run with a daemon option to reserve values to identify token cards
  - Authentication-Type attribute values are automatically reserved to identify token cards
  - Ascend NavisRadius does not reserve Password attribute values
- AppleTalk Remote Authentication (ARA)
  - Ascend NavisRadius uses a new Authentication-Type attribute value to add ARA authentication to a user profile
  - Ascend NavisRadius uses a new value to indicate ARA is a user's `Framed-Protocol`
- Identifying Network Access Servers (NAS) Vendors
  - Ascend NavisRadius provides a means of identifying the vendor who manufactures a NAS
  - Based on the vendor identification, Ascend NavisRadius only sends appropriate vendor-specific attributes to the NAS
- New attribute names
  - Ascend NavisRadius supports attribute names as they appear in the IETF RADIUS RFC, not the RADIUS draft

Operational differences

- Reading changes in data files
  - Ascend NavisRadius does not re-read the files when a change is made in the `clients` or `users` files
- Ascend NavisRadius supports new daemon command line options
  - `-C` allows token-caching

  - `-f` changes the default finite state machine file which monitors the status of authentication requests
  - `-g` changes the default location where the log `radiusd` activities is sent
  - `-h` activates help
  - `-k` provides key information
  - `-P` allows password changes
  - `-p` changes the authentication port (this supports the standard, but is different than the Ascend free RADIUS
  - `-pp` changes the default relay authentication port
  - `-q` changes the default accounting port
  - `-qq` changes the default relay accounting port
  - `-r` changes the first default separator in a realm-aligned user's user name
  - `-rr` changes the second default separator in a realm-aligned user's user name
  - `-t` sets the timeout limit for the authentication request
  - `-z` removes old `logfile` at startup
- Debugging
  - Ascend NavisRadius automatically sends debugging information to a file called `radius.debug` instead of to `stderr`

User's Guide summary

This list summarizes the information in chapters 2 through 8 of the Ascend NavisRadius User Guide and the appendixes of the Ascend NavisRadius References. Chapters 2 through 4 constitute a quick start guide for experienced RADIUS users.

Ascend NavisRadius User's Guide

Chapter 2, "Quick Setup," is the first of three chapters written for users who already understand RADIUS authentication and authorization and want to start using the NavisRadius graphical interfaces. It describes how to plan for and set up NavisRadius. It also lists the RADIUS daemon options that control the location of the NavisRadius data directory, the server's authentication and accounting ports, and the depth of information collected when debugging is turned on.

Chapter 3, "Installing Ascend NavisRadius,"describes how you install demonstration copies of NavisRadius and obtain a temporary 30 day license key. It also discusses permanent installation of NavisRadius.

Chapter 4, "Configuring NavisRadius Files." describes the configuration of the users, clients and authfile data files. users files contain user profiles.

Chapter 5, "Understanding NavisRadius,"explains how NavisRadius communicates with Network Access Servers (NAS), which initiate authentication requests. The explanation describes RADIUS attributes and values, RADIUS accounting and the NavisRadius data files.

Chapter 6, "Creating Example Client, User and Realm entries," is another tutorial explaining the format of the users, clients, and authfile files. The focus of this chapter is manual editing of the data files to create clients, users and realms. The chapter also explains NavisRadius utilities.

Chapter 7, "Using the runtime SQL Database," explains NavisRadius support for an SQL compliant database.

Chapter 8, "Configuring advanced features," explains NavisRadius support for advanced features.

Ascend NavisRadius References

Appendix A, “Ascend NavisRadius files and commands,” contains information on the basic files and commands.

Appendix B, “iPass notes.” discusses iPass and how to use it with NavisRadius.

Appendix C, “Attributes Reference,” is a list of NavisRadius attributes. In addition to RADIUS attributes, the list includes vendor-specific attributes developed by Ascend Communications.

Appendix D, “SQL script for authentication and accounting table,” discusses SQL script for creating a DBMS called authentication and accounting.

Appendix E, “radodbc.map file,”Appendix E, “radodbc.map file,” provides a cross reference to the SQL scripts for authentication and accounting tables.

Appendix F, “Ascend NAS Accounting codes,” is a collection of tables that explain the accounting codes supported and used by the Ascend MAX family of routers.

Quick Setup

2

This chapter begins a quick start guide that includes Chapters 3 and 4. Together, they provide information experienced RADIUS users need to install and to use Ascend NavisRadius.

Planning to set up Ascend NavisRadius

Following are preparations you should take to set up NavisRadius.

1 Review the NavisRadius attributes.

Appendix C, “Attributes Reference,” contains descriptions and values of Ascend-specific and RADIUS authentication, authorization and accounting attributes. NavisRadius’s `dictionary` file is also a list of the attributes.

2 Gather information about the Network Access Servers (NAS) and other authentication servers that will be your NavisRadius server’s clients.

3 Gather information about individual users that you will enter in the `users` file’s profiles.

Steps 2 and 3 are the most time consuming. You might be able to collect user information from an existing user database. If your NAS clients are Ascend units, you might find useful information in the units’ Answer and Connection profiles.

4 Decide if the users NavisRadius authenticates should be separated into realms.

5 Decide where to install NavisRadius on the workstation.

You can create a directory for the program, or accept the installation script’s default location. The default is `/etc/raddb` on UNIX machines and `C:\Ascend\NavisRadius` on Windows NT machines. The directory and other default settings, such as the RADIUS authentication and accounting ports, work for most installations, but you can change them using the `radiusd` command line options listed in Table 2-1.

6 Read the Ascend NavisRadius `README` file.

Information that differs from the steps outlined here, and elsewhere in the Ascend NavisRadius User Guide, would be included in a `README` file.

Default Ascend NavisRadius settings

`radiusd` is the RADIUS daemon. A discussion of `radiusd` is in Appendix A, “Ascend NavisRadius files and commands,” lists the daemon command line options and their default settings. You can change the default settings.

Table 2-1. RADIUS daemon command line options

Option	Description	Default
`-a`	Accounting directory Location of accounting detail file	`/etc/radacct`
`-C`	Allow token caching	Not enabled
`-cwd`	Current working directory Only affects file system operation for files with no relative file names	`/etc/raddb`
`-d`	Data directory Location of `dictionary`, `clients, authfile`, and `users` files	`/etc/raddb`
`-f`	Finite State Machine (FSM) file Table describing machine state	`radius.fsm`
`-g`	Logging Style for logging warning, error and informational messages	Enabled. Output to `logfile`
`-h`	Help messages Sends help messages to standard output	Not enabled

Table 2-1. RADIUS daemon command line options

Option	Description	Default
`-i`	Allow i-Pass authentication	Disable ipass authentication
`-noservice`	Do not run radiusd as a service. This option is available only on Windows NT	Run as a service
`-oa`	Allow ODBC accounting	Disable ODBC accounting.
`-p`	Authentication port Port receiving authentication requests	1812
`-P`	Allow password change messages	Disable password change messages
`-q`	Accounting port Port receiving accounting information	1813
`-pp`	Alternate authentication relay port Port for relaying request to proxy server	1812 (same as authentication port)
`-qq`	Alternate accounting relay port Port for relaying information to proxy server	1813 (same as accounting port)
`-s`	Ascend NavisRadius processing mode How Ascend NavisRadius handles each request	Multi-thread process

Table 2-1. RADIUS daemon command line options

Option	Description	Default
`-t`	Timeout Time server will wait during inactivity before timing out	Fifteen (15) minutes
`-v`	Display Ascend NavisRadius version number to standard output	Not enabled
`-x`	Debugging Level of debugging output When enabled, `radius.debug` is default file to receive information	Not enabled
`-z`	Empty logfile and debug file Not applicable if debugging is not enabled or if syslog is used for logging.	Not enabled

Installing Ascend NavisRadius 3

This chapter describes installation of the NavisRadius software and file configuration.

Note: *A license key is required.*You must obtain a license key from Ascend to use Ascend NavisRadius.

Who should read this chapter

This chapter provides information about installation of the NavisRadius software on supported platforms: HP-UX, Solaris, and NT.

You should be familiar with RADIUS if you plan to install and configure NavisRadius before reading Chapter 5, “Understanding NavisRadius,” and Chapter 6, “Creating Example Client, User and Realm entries,”. Some NavisRadius features discussed in chapters 3 and 4, such as realms, go beyond standard RADIUS. The concept of user realms and the configuration of Ascend NavisRadius’s `authfile` file should not be difficult for experienced RADIUS administrators.

Technical Support

Contact the Ascend Technical Support Center at 1-800-ASCEND4 if you need help. The center is open Monday through Friday 6:00AM to 6:00PM PST.

Pre-installation considerations

Prepare for Ascend NavisRadius installation by reviewing the lists in the following two subsections. In addition, be aware that *you need root privilege on the NavisRadius workstation when you install the program.* You do not need root privilege to change NavisRadius’s default configuration after installation.

What you need before you start

To install and activate Ascend NavisRadius you need the following:

- Ascend NavisRadius CD ROM.
- Ascend NavisRadius license key.
- To get a license key, fax the NavisRadius license request form to the following number :
  - (516-468-4889 (in the United States),
    (001-516-468-4889 (outside the United States.)

Please ensure that you provide all relevant information requested for the license key to be created and emailed or faxed back to you.

- Workstation running one of the following operating systems:
  - Solaris 2.5.1 (SPARC)
  - Windows NT 4.0 Service Pack 3 (X86)
  - HP/UX 10.20 (PA-RISC)
- TCP/IP connection between the Ascend NavisRadius server and Network Access Server (NAS) clients.

Clients and Users files configuration

To configure entries in the `clients` and `users` files, you need the:

- Clients' secret keys.
- Users' authentication and authorization information.

Ascend NavisRadius CD-ROM

The Ascend NavisRadius CD-Rom contains the program's binary files, documentation, installation scripts, and help files.

Use a text editor to open the README file on the Ascend NavisRadius CD-ROM. It contains instructions for installing Ascend NavisRadius .

Ascend NavisRadius CD-ROM directories

Ascend NavisRadius CD-ROM's alphabetically arranged directories contain:

- Adobe Acrobat Reader
- Documentation for Ascend NavisRadius in Acrobat Reader (.pdf) and Postscript (.ps) formats
- The Ascend NavisRadius binary files
- iPass Roamserver software (not available on Windows/NT)

Adobe Acrobat Reader

All user documentation on the Ascend NavisRadius CD-ROM are provided in Adobe Acrobat PDF format. The CD-ROM contains the Acrobat Reader installer, which you can use to install Acrobat Reader. Run Acrobat Reader to open the PDF documents.

Install Acrobat Reader for Windows by double-clicking `ar32e30.exe`. Follow the directions for your operating system in the corresponding operating system directory.

Documentation

The Documentation directory contains postscript and PDF versions of the Ascend NavisRadius User Guide.

Note: For ease of installation, the description of each system directory is followed by the installation procedure for that system.

HPUX directory

The `hpux` directory contains the binaries for Ascend NavisRadius. The software may be installed from the CD-ROM by following the instructions in the `Readme` file in the CD-ROM's base directory. The `Readme` file has information about this version of NavisRadius and your preparations for installing the program.

You can use the `swinstall` command to install NavisRadius for HPUX 10.x. This command enables you to automatically or interactively install software. If you choose to interact with the installation process, you can do so through a terminal interface or a Graphical User Interface. The `swinstall` command and its options are described in detail in the HP-UX 10.x `swinstall` man page.

You will need the system's hostname to produce an NavisRadius license key. You must enter your key in a file called `licenses` which needs to be placed in the NavisRadius data directory.

You may select from two packages for installation: ASNDac - NavisRadius, and ASNDipass - NavisRadius ipass support.

Installing NavisRadius on HP-UX

To install NavisRadius with `swinstall`, do the following:

1. Issue the command `swinstall`
2. Specify the path to hpux10-hppa.tar

You can now proceed to customize your installation.

⚠ **Caution:** The HP/UX 10.20 needs the patches PHCO_13626, PH KL_14282, PHSS_15043 to properly install and run NavisRadius.

Note: If you are installling NavisRadius on HP-UX from the Web, you need to uncompress, but not untar the `hpux10-hppa.tar.z` file

Using swinstall to replace an earlier version of NavisRadius

swinstall command

By default, swinstall supports updates to higher revisions of software. The files of the lower version may be overwritten or temporarily saved, depending on the argument for the swinstall autorecover-product option. swinstall moves files of the lower version of the software which are obsolete, but are not overwritten.

Uninstalling NavisRadius on HP-UX

You can remove NavisRadius with the `swremove` command. This command and its options mirror the swinstall command and its options. The HP-UX swremove man page explains the swremove command. At the prompt, enter:

- `swremove`

Solaris directory

The `solaris` directory contains the binaries for Ascend NavisRadius. The software may be installed from the CD-ROM by following the instructions in the `Readme` file in the CD-ROM's base directory. The `Readme` file contains information about this version of Ascend NavisRadius and your preparations for installing the program.

You will need the system's hostname to produce an Ascend NavisRadius license key. You must enter your key in a file called `licenses` which needs to be placed in the Ascend NavisRadius data directory.

You may select from two packages for installation: ASNDac - NavisRadius, and ASNDipass - NavisRadius ipass support.

Installing NavisRadius on Solaris

You can use the `pkgadd` command to install Ascend NavisRadius on Solaris machines. You can also install it using the admintool.

To install NavisRadius using `pkgadd:`

- Issue the command `pkgadd -d <`path to Solaris on the CD`>`

Or

1 Start the Admintool

2 Select Software from the Browse menu

3 Select Add from the Edit menu

4 Specify path to Solaris on the CD

Note: If you are installing NavisRadius on Solaris from the Web, you need to uncompress and untar the `solaris25-sparc.tar.z` file.

Uninstalling NavisRadius on Solaris

- Issue the command `pkgrm ASNDac` or `ASNDipass` (for iPass)

Or

1 Start the Admintool

2 Select Software from the Browse menu

3 Select Delete from the Edit menu

Windows NT 4.0 directory

The `Winnt` directory contains the binaries for Ascend NavisRadius. The software may be installed from the CD-ROM by following the instructions in the `Readme` file in the CD-ROM's base directory. The `Readme` file contains information about this version of Ascend NavisRadius and your preparations for installing the program.

You can install and run Ascend NavisRadius on machines running the Windows/NT 4.0 (Service pack 3) operating system. NavisRadius for Windows/NT performs all the currently supported NavisRadius authentication, authorization and accounting functions. Please note that iPass support is not available on NT.

You can install Ascend NavisRadius for Windows 4.0 from the CD-ROM's `Setup.exe` file. It begins a utility which guides you through the Ascend NavisRadius installation. Setup creates system directories and extracts the program's files automatically.

Setup installs Ascend NavisRadius in the default directory, `C:\Ascend\NavisRadius\` if you don't enter an alternative.

You will need the system's hostname to produce an Ascend NavisRadius license key. You must enter your key in a file called `licenses` which needs to be placed in the Ascend NavisRadius data directory.

You need to create and place the `licenses` file in the `C:\Ascend\NavisRadius\raddb` directory.

RADIUS service

NavisRadius for Windows/NT4.0 runs as a Windows/NT service. Services can be configured via the Control Panel interface, or the Admin utility, a graphic interface installed with NavisRadius. By default, the NavisRadius service is set to run manually. You can use Control Panel's services option to configure NavisRadius to start automatically when you start Windows/NT 4.0.

Logging

Information about NavisRadius for Windows/NT 4.0 is captured in three different places on the system: the Windows/NTevent log and two files, logfile.yymmdd.txt and radius.debug.txt.

Directories and Files

Following are the subdirectories and the files the Install Shield wizard creates under the directory, `C:\Ascend\NavisRadius`:

`\Accounting`

- The `\Accounting` subdirectory is empty.

`\Bin`

- The `\Bin` subdirectory contains the `AC_Admin.exe` and `radiusd.exe` files.

  These are the Ascend NavisRadius administration program and the RADIUS daemon, respectively. `AC_Admin` enables you to start, stop, refresh, and query *RadiusSrv,* the Ascend NavisRadius server.

`\raddb`

- The `\raddb` subdirectory contains the `authfile`, `clients`, `dictionary, users`, and `vendors templates` files.

Installing NavisRadius on Windows/NT

Following are steps for installing NavisRadius for Windows/NT 4.0. Use the `setup.exe` to install NavisRadius.

If you choose the default installation parameters, the utility creates the C:\Ascend\NavisRadius directory and the Ascend program group if they do not exist. The utility also creates four subdirectories under C:\Ascend\NavisRadius: \Accounting,\bin, \Database, and \Examples.

To install:

1 From the CD point to the `Winnt` directory.

2 Run the Setup.exe.

3 Answer all the questions with which you are prompted.

4 If this is the first time the RADIUS service has been installed, reboot the system so it will recognize the NavisRadius RADIUS service.

The service appears in the system's list of services as RadiusSrv.

Uninstalling NavisRadius NT

You can remove NavisRadius with the Control Panel Add/Remove utility. Configuring NavisRadius NT as a service You can configure the NavisRadius RADIUS server to run automatically when you start the system.

1 Click Start on the task bar.

2 Click Settings and select Control Panel.

3 Click the Services icon.

4 Select RadiusSrv.

5 Click Startup.

6 Click Automatic

NavisRadius Configuration

Configuring the Ascend NavisRadius Service

To configure the Ascend NavisRadius service on Windows NT:

1 Click *Start* on the taskbar.

2 Click *Settings* and choose *Control Panel*.

3 Select the *Services* icon from the *Control Panel* icons.

A list of available services appears in a dialogue box Changing Ascend NavisRadius's default configuration.

Changing Ascend NavisRadius' default configuration

By default, the Ascend NavisRadius server, called *RadiusSrv* in the list of Windows NT services, is a Manual service. The alternative designation is Automatic service. An Automatic service starts when the server boots up. You can change RadiusSrv to an Automatic service:

1 Click *Startup*...

2 Click *Automatic*.

Changing Ascend NavisRadius parameters

You can also change the options of the RADIUS daemon by typing new options and values in the Startup Parameters textbox in the Services dialogue box. For example, by default, the RADIUS daemon listens for authentication requests at port 1812. If you want to change the port number to 5625, type `-p 5625` in the Startup Parameter textbox. You can change as many different RADIUS daemon options and values as you want in the textbox.

If you want to change a default or current Ascend NavisRadius path name, you must type two backslashes between each segment of the path name. The first backslash is an escape character. For example, to change the data directory from `C:\Ascend\NavisRadius\raddb C:\Ascend\NR\raddb`, type the following in the parameters textbox:

```
-d C:\\Ascend\\NR\\raddb
```

Configuring NavisRadius Files

4

This chapter introduces and describes Ascend NavisRadius' `users`, `authfile` and `clients` files.

The configuration process

The information you gather as you prepare to configure Ascend NavisRadius should indicate whether you will benefit by grouping users in realms. Follow all of the steps in “Configuration steps,” below, if you have determined that realms are beneficial for authenticating and authorizing your user community. If you determine that user realms are unnecessary, skip step 2 “Add realms in authfile”.

Although the order of the configuration steps is arbitrary, users cannot be defined by their realms if you have not created a list of realms they can belong to. Also NavisRadius checks the users file before checking the authfile, even if users are grouped in realms.

Many system administrators do not enter individual user profiles in the users file. They only create a DEFAULT user in the users file and a DEFAULT realm in the authfile. The first line they create for the DEFAULT user profile contains the attributes Password and Authentication-Type. They enter UNIX-PW as the Password value and Realm as the Authentication-Type value. In the DEFAULT realm entry in the authfile they enter File as the value for the Realm/DNS/File field and a name in the Prefix field. These actions:

- Prevent people with access to the users file from reading user passwords because the passwords are stored in the server’s /etc/passwd file.
- Similarly protect user authentication and authorization attributes because they are not stored in the users file, but in a file named *.users, where * is the name entered in the DEFAULT realm’s Prefix field.

Configuration steps

Following, are the steps to use to configure NavisRadius:

1 Add a DEFAULT user in the users file.

2 Add realms in authfile, including a DEFAULT realm.

3 Add clients in clients file.

4 Add users in users file.

Understanding users files

A `users` file is a list of user profiles containing attribute/value pairs. The people the profiles define, are users NavisRadius can authenticate for a NAS or a server listed in the `clients` file.

You may create one `users` file or many `users` files for the NavisRadius server. If you organize users into realms, you can create separate `users` files for each realm. For example, you might create a file named `ournet.users` to contain the profiles of users associated with a realm called *Ournet*. A prefix such as *ournet* is derived from the value of the Realm/DNS/File field in an `authfile` realm entry. The realm's Type field value must be *File* to enter a name in the Realm/DNS/File field. See "authfile" on page A-2 for more information about the fields in `authfile.`

Users file attributes

You can only enter in a `users` file the attributes that appear in the `dictionary` file. User profile attribute/value pairs are classified as either check-items or reply-items.

Check-items are attribute/value pairs NavisRadius compares to the attributes/values it receives from a client to authenticate a user.

Reply items are attribute/value pairs NavisRadius sends the client to authorize a link and services if authentication is successful.

If authentication fails, NavisRadius typically sends an attribute/value containing a message about the authentication failure.

Users file format

The `users` file is installed in the `/etc/raddb` directory. However, you may reconfigure NavisRadius to use a different directory. (For more information on changing NavisRadius defaults, see "Default Ascend NavisRadius settings" on page 2-3.)

A `users` file may contain general comments about the file and separate comments about individual user's profiles. Each user profile contains one line of check-items for authentication and, possibly, one or more lines of reply-items for authorization. A view of the users file, shows the format of the entries:

```
#Comment

username check-item [, check-item]...

   reply-item,
   reply-item
```

Example users file entry

A `users` file entry viewed with a text editor looks like this:

```
#This user began it all.

gwash@whitehse Password=paterusa

   Service-Type=Framed,
   Framed-Protocol=PPP,
   Framed-IP-Address=105.23.0.1,
   Framed-IP-Netmask=255.255.255.0
```

Entering a connection configuration attribute more than once in a user profile

It is possible, and in some cases necessary, to enter the same connection-configuration attribute more than once in a user profile. If this occurs by accident, the Network Access Server that receives the attributes should still establish a connection for an authenticated user. The NAS will set up the connection using the last value it reads for the duplicated attribute.

For example, suppose you inadvertently include the attribute Service-Type twice in the same user profile. You enter `Framed` as the value of the first entry and login as the value of the second entry. The NAS receives NavisRadius' Access-Response packet containing these attribute/value pairs and reads Service-Type=Framed then Service-Type=Login. The NAS will set up a login connection rather than a framed connection because Login is the last value it receives for the attribute Service-Type.

An example of an attribute that should appear more than once in a user profile is the Ascend-Menu-Item attribute. You can use this attribute to create a text message on an authenticated user's screen. The message generally provides a menu from which the user may select one of several options.

User profile format example

This is an example of an entry for a dial-in user. Its format is the format of a user profile in a users file or prefix.users file.

```
Dial-in-User Password = “XXX”
   Service-Type = Framed,
   Framed Protocol = MPP,
   Framed-IP-Address  = 0.0.0.0,
   Framed-IP-Netmask  = 255.255.255.0,
   Ascend-Link-Compression  =  Link-Comp-Stac,
   Ascend-Idle-Limit  = 00
   Framed-Compression  = Van-Jacobson-TCP-IP
```

Understanding the authfile

The `authfile` is a list of the realms NavisRadius recognizes. Generally, a realm is a group of users who share a common characteristic, such as being customers of the same Internet Service Provider. Realm members' user profiles can be nearly identical, perhaps varying only in the values of their User-Name and Password attributes. Members of realms enter their user names in the format *username@realm*.

When one NavisRadius server supplies authentication, authorization and configuration services for multiple users, you can effectively control access to user profile information by limiting the viewing and editing privileges in each realm to the realm's `users` file.

NavisRadius's realm feature is *the* solution when you should isolate different groups of users from one another on one NavisRadius server. The feature is not, however, the recommended solution for every NavisRadius environment. Many security policies specifically state that all NavisRadius user profiles must be stored in one centralized `users` file.

authfile format

Create an `authfile` file in the NavisRadius data-directory, `/etc/raddb,` if you want to use the NavisRadius realm feature. `authfile` is not created for you by the Install script or the pkgadd program, but a file containing examples of authfile entries is. You can copy examples of the entries from the `authfile.ex` file to the authfile file you create. authfile contains one line of information for each realm.

Each entry in the `authfile` must include values for `authfile`'s Realm-name and the Type fields. Each entry can also include optional values. Most values you can enter for the type of authentication are incomplete without a reference to the location of a host or file, so you might also be required to provide a Kerberos realm name, a DNS hostname, or a filename in the `authfile` entry.

Following are the authfile's required and optional fields:

```
Realm-name [( Alias[, Alias])] [-Protocol] Type
[Realm/DNS/File]
```

For example, an authfile could contain the following two lines:

```
umich.edu (wolverines) -pw afs-krb umich.edu mich
ohio.org  (buckeye, bucks) file buckeye
```

Understanding the clients file

The `clients` file is a list of all the Network Access Servers (NAS) and remote NavisRadius servers that can send Access-Requests to the NavisRadius server. If a client is not represented by an entry, the server discards its messages.

Clients file format

You must create the `clients` file. Like authfile and users file, `clients` resides in the `/etc/raddb` directory. Copy examples of clients file entries from `clients.ex,` which is copied on the server when NavisRadius is installed. Unlike authfile, it is not optional and you must not delete it. Entries in clients file may include five fields. Two, System-name and Key, are required.

System-name has an optional component called Port. The Port component overrides NavisRadius's default authentication port number. The five fields, with example entries are:

System-name:*Port*	Key	[*Type*]	[*Version*]	[*Prefix*]
marlowe:1945	loring	type=nas	v2	mystery.

Caution: The Prefix field entry in the clients file must contain a trailing period (.). If you enter the previous example in your clients file, NavisRadius searches for realms associated with the NAS named Marlowe in a file named `mystery.authfile`. If your prefix entry does not include a trailing period, NavisRadius searches for a file named `mysteryauthfile.`"Prefix field (option)" on page A-10 also discusses the format of the `clients` file Prefix field.

Note: Although the function of an `authfile's` Realm/DNS/File field entry can be similar to a `clients` file's Prefix entry, the `authfile's` Realm/DNS/File field's entry does not require a trailing period. When the `authfile` Type field value is `file,` enter a name in the Realm/DNS/File field. The name is a prefix for a `users` file. The values `file` and *name* in the Type and Realm/DNS/File fields indicate that the profiles of a realm's users are stored in a file called *name*.`users`. NavisRadius assumes that the Realm/DNS/File field entry includes an implied trailing period. "authfile" on page A-2 and "Other users files" on page A-46 contain more information about `authfile` entries and Realm/DNS/File.

Understanding NavisRadius

5

This chapter is an overview of the Ascend NavisRadius features for authentication, authorization and accounting functions and also discusses support for alternative authentication methods

Who should read this chapter

This chapter provides information not included in the three quick start chapters, which are for people who have some understanding of the way that Ascend NavisRadius or RADIUS authentication, authorization and accounting services work. Read this chapter if you need some background information before plunging into creating user profiles, identifying clients, and configuring realms.

The chapter begins with brief overviews of NavisRadius's authentication, authorization and accounting services, and moves to descriptions of the program's foundations: the client/server messages and the field entries of each NavisRadius file. It closes with information about NavisRadius's support of proxy RADIUS and other authentication methods. This approach seeks to illustrate the way that many small pieces of data are collected to support NavisRadius's services, and the way those services can be joined with other products to implement AAA services.

Ascend NavisRadius overview

NavisRadius authenticates and authorizes remote users and, optionally, stores accounting information about the users' network connections.

NavisRadius authentication and authorization go hand-in-hand. They are conducted via four types of Access messages that pass between a Network Access Server (NAS) and an NavisRadius server. The messages, Access-Requests, Access-Responses, Access-Challenges and Access-Rejects, are composed of User Datagram Protocol (UDP) packets.

NavisRadius accounting is optional. When you make accounting operational, the NAS sends Accounting-Request messages to NavisRadius that include details about a user's network connection. NavisRadius acknowledges receipt of the NAS messages with Accounting-Response messages.

For explanations of NavisRadius and RADIUS Accounting messages, see "Understanding RADIUS messages" on page 5-9.

How Ascend NavisRadius works

Although simplified, the following steps explain how NavisRadius works:

1 An NAS, also referred to as a client, contacts the NavisRadius server and asks NavisRadius to verify the identity of a user who has requested network access.

2 NavisRadius searched its client-list file for a match with the client-identification information in the client's message. The information includes a secret key, on which NavisRadius performs an MD-5 checksum. If the result is valid, NavisRadius extracts user-identification information from the message.

3 NavisRadius searches a file of user profiles. If it finds a profile that matches the user information sent by the client, NavisRadius sends the client a user-authentication message.

4 The message verifies the user's identity, specifies the extent of the user's access to the network and to network services, and defines how the user's link to the network is to be configured.

Step 3 includes additional operations when the user's profile specifies one of the following conditions:

- User's profile is on another RADIUS server.

  NavisRadius becomes a proxy for the RADIUS server.
- Another form of authentication, such as Kerberos or TACACS is required.

  NavisRadius opens an authentication session with the appropriate server.
- Authentication requires a SecureID or other token key.
- NavisRadius send's the user's token-key information to the token card server.

Ascend NavisRadius files

To perform this way, NavisRadius needs access to a lot of information, including:

- who its clients and their users are
- what its clients and their users will provide as identifying information
- where to find matching information to identify clients and their users

- how its clients should configure their users’ access and links

If you read the Chapter 1, “Introducing Ascend NavisRadius,” you remember the example “story” about a telephone call from a building inspector who wants to visit a construction site. The story has two purposes. One purpose is to illustrate the many types of information you may collect for authentication and authorization. The second purpose is to relate the types of information to typical user attributes found in the NavisRadius `dictionary`. If you skipped the introduction, here is a list of the information collected during the call and the corresponding attributes from the NavisRadius `dictionary` file.

Table 5-1. Converting information to Ascend NavisRadius attributes:

Information	**Attribute**
Caller’s name	User-Name
Caller’s means of identification	Password
Caller’s location	Framed-IP-Address
Caller’s telephone number	Calling-Station-ID
Your means of verifying the caller’s TN	Ascend-Callback
Caller’s requested destination	NAS-IP-Address
Caller’s means of travel	Framed-Protocol
Activity for which caller requests approval	Service-Type
Your means of controlling caller’s activities	Ascend-Data-Filter
Your means of restricting the length of caller’s approved access	Expiration
Your means of keeping track of the caller’s visits	RADIUS Accounting

After obtaining the information you can enter it in a file, or files, that reside in the NavisRadius data directory. The files are named `users`, or `prefix.users`, where prefix is the name of a realm which defines a group of users.

The data directory also contains other NavisRadius files named `dictionary, authfile` and `clients.` NavisRadius consults these files when performing the steps outlined in "How Ascend NavisRadius works" on page 5-3. The `clients` file, for example, is a list of all clients that NavisRadius can authenticate. `authfile` is a list of the user realms where NavisRadius stores the *prefix* in `prefix.users.` Each of these files is described in more detail later in this chapter.

Types of file entries

Ascend NavisRadius stores data file information as field values or attribute/value pairs. The format of the data depends on the data file.

You enter field values in the `authfile` and `clients` files. You must enter a value for some fields in each file because NavisRadius cannot parse the entry, or define the entry's realm or client, if the field is blank. Entering values in the other fields is optional. NavisRadius can parse the file entry and recognize what it describes if you leave these fields blank. Realm-Name and Type are required `authfile` fields. Version and Prefix are optional `clients` file fields. (The status of each `authfile` and `clients` field is discussed in Appendix A, "Ascend NavisRadius files and commands."

You enter values that define attributes in all types of users files, including templates. The difference between an attribute and a field is slight, but significant. Attributes describe the users NavisRadius authenticates and the way the users can connect to other machines. A few attribute values are required. For example, NavisRadius must find the user's identity and an authentication method in the user profile. It is misleading to describe the rest of the attributes as optional, because user profiles almost always contain some authorization attribute/values pairs. NavisRadius sends the authorization attribute/values, which describe the user's connection configuration, to the NAS.

Understanding authentication

NavisRadius' primary function is to confirm that someone requesting a connection to or from the network has permission to make the connection and to use network services. NavisRadius authenticates users by comparing the user's name and the password with information in its own files. The user's name and address are sent by a NAS or a proxy server. The process that defines the messages that pass between the NAS and NavisRadius includes the following:

1 A NAS or proxy server sends an Access-Request message to the NavisRadius server. The message includes the names of the user and the NAS and the user's password. To learn more about NavisRadius proxy servers see "Proxy Ascend NavisRadius servers" on page 5-29.

NavisRadius compares information in the Access-Request message with data in the `clients` file and a `users` file. The `clients` file contains the names of all the NAS units that may request user authentication from the NavisRadius server. NavisRadius `users` files are databases containing information about legitimate users NavisRadius can authenticate. The value of a `clients` file field named Key is a secret shared by NavisRadius and the NAS, which they use to verify each other's identity. A `users` file stores information in user profiles containing attribute/value pairs. NavisRadius's key for finding a user profile in the `users` file is the value of the User-Name attribute.

Note: NavisRadius can utilize more than one `users` file. The user's name determines which of the `users` files NavisRadius consults when trying to match the information sent by the NAS.

2 NavisRadius sends the NAS an Access-Response message if the data in the Access-Request message matches the NAS information in the `clients` file and the user information in the appropriate `users` file.

or

NavisRadius sends the NAS an Access-Reject message if the name of the NAS is not in the `clients` file or the user's name or password does not match any corresponding entries in the `users` file.

Ascend NavisRadius authentication options

NavisRadius supports a variety of authentication methods. “NavisRadius authentication methods” on page 5-27, explains NavisRadius support for:

- PAP
- CHAP
- S/Key
- RADIUS proxy servers
- Kerberos
- TACACS and TACACS+
- Token Keys

Understanding authorization

NavisRadius’ second function is to define the services that users may access and the connections they can make. NavisRadius does this by using the information it stores in a `users` file. NavisRadius might answer a NAS request by sending an Access-Response packet authorizing the NAS to allow the user to connect to a specific network machine, such as a Telnet server.

Authorization information that NavisRadius sends to a NAS might also tell the NAS how users are allowed to make internal and external connections. For example, NavisRadius’s authorization could instruct the NAS to allow an incoming user’s connection to a local Telnet server via an unframed protocol, but force another, outgoing, user to connect remotely with framed Point-to-Point Protocol.

Attribute/value pairs define the specific authorizations granted a user. The `dictionary` file explicitly lists NavisRadius’s attributes and their possible values. Some NavisRadius attributes come from the RADIUS standard, but most are Ascend’s vendor-specific additions. The RADIUS protocol includes a means of adding proprietary attributes. Vendor-specific attributes are generally associated with the capabilities of a particular NAS unit.

Understanding accounting

NavisRadius' third function is to serve as a repository of information about network connections. When it has authenticated a user and authorized the connection and services, NavisRadius might receive an accounting of the connection from the NAS. The accounting can include such things as the start and stop times of the connection and the number of packets and octets transmitted over the connection. This information can be valuable for evaluating security policies and network usage.

RADIUS accounting has its own set of attributes and, as with authorization attributes, provides more accounting attributes than those described in the RADIUS protocol. Accounting attributes and their values are also explicitly listed in the NavisRadius `dictionary` file.

If a NAS is configured for RADIUS accounting, it sends an Accounting-Request message that includes an Accounting-Start packet to NavisRadius which responds with an acknowledgment that the start packet was received. At the end of the accounting delivery, the NAS sends another Accounting-Request message that includes an Accounting-Stop packet. NavisRadius also acknowledges this packet. NavisRadius can also be a proxy accounting server, transmitting the NAS's Accounting-Requests to another RADIUS accounting server.

There is no default limitation on the number of times the NAS may send Accounting-Request messages in the event that it receives no acknowledgment. You can specify a time frame the NAS will wait to receive an acknowledgment or limit the number of times the NAS can send an Accounting-Request, or both

Some NAS units, such as the Ascend MAX, MAX TNT, can be configured to send accounting information to more than one accounting server. This provides the NAS with an alternative if its primary accounting server is down or unreachable.

Understanding RADIUS messages

This section begins a discussion of some facets of the RADIUS protocol. RADIUS is a foundation for NavisRadius services, although the program extends those services and supports many features that are not found in the RADIUS protocol. The RADIUS protocol is defined in RFC 2138. NavisRadius passes messages between its servers and clients via the method outlined in that RFC. Like RADIUS, it also stores attribute/value pairs in files and packages them within the server's messages.

RADIUS packets

RADIUS is a User Datagram Protocol (UDP). Sending UDP packets rather than Transmission Control Protocol (TCP) packets, reduces the length of time the user must wait for authentication. To understand why, consider the communications that pass between the client and the server while the user waits.

Suppose that the NAS sends the server an Access-Request message that is garbled in transmission. In plain language the message the NAS, or another server acting as a client, sends is:

> "Here's some information about a user who wants me to set up a connection for him. Please authenticate this user for me. I'm a client of yours and here's the secret key that identifies me."

The garbled message the RADIUS server receives might be:

> "Here's some ...to set up a connection for him. Please ...this user for me. I'm a ... and here's the secret you can use to identify me."

This message doesn't make sense to the server, which has not received all the information it needs. Therefore it doesn't respond to the client. After a specified time, the client sends the Access-Request message again and it arrives intact.

The server compares the secret key it receives to one it has on file for the client. If the secret keys don't match, the server might send an Access-Reject packet to the client explaining why it can't authenticate the user. If the secrets do match, the server responds in one of two ways:

1 The server sends an Access-Accept packet containing this message:

“Yes, I can verify the identity of the user. Here is some information about the way to set up the user’s connection.”

2 The server sends an Access-Challenge packet containing this message:

“Yes, I will try to verify the user’s identity. Ask him to respond to this challenge.”

The messages are so short the user doesn’t know the Access-Request was transmitted twice. UDP doesn’t add time to the exchange because the protocol does not require that the client and server negotiate how they will handle the connection. The client and server accept each other’s messages with very little concern for transmissions errors.

TCP, the alternative to UDP, is connection-oriented and it does provide end-to-end error checking. If handled by TCP, the exchange described above would take longer because TCP requires negotiations and error-checking.

Types of RADIUS messages

The example client/server exchange in “RADIUS packets,” above, identifies four types of RADIUS messages:

- Access-Request
- Access-Reject
- Access-Accept
- Access-Challenge

If you employ RADIUS Accounting, the client and server might also exchange the following types of messages:

- Accounting-Request
- Accounting-Response

The information in each RADIUS message is encapsulated in a UDP packet’s data. A packet is a block of data set in a standard format for transmission. It is accompanied by other information, such as the origin and destination of the data. By default the server’s source and destination for a RADIUS packet is port 1812. The client may not have a default port for exchanging the packets, but may be configured according to the documentation that accompanies the unit. You may override the server’s default port from the command line by starting the RADIUS daemon with the -p option. For a description of this daemon option, see “radiusd,” in Appendix A.

Each RADIUS packet contains five fields measured in octets, each of which is an eight-bit chunk of data representing an ASCII number or letter. The number of octets in the Attribute field is variable. The Authenticator field contains sixteen octets, and the other fields consist of one or two octets. The total number of octets in the packet is between 20 and 4096. Table 5-2 describes the five fields.

Table 5-2. RADIUS packet fields

Fields	**Description**
Code	One octet that identifies the message by type
Identifier	One octet used to match a client’s Access-Requests and a server’s replies
Length	Two octets representing the number of octets in the entire packet
Authenticator	Sixteen octets representing a value used to authenticate the server’s reply
Attribute(s)	Variable number of octets representing the numerical equivalents assigned to attributes and their values

RADIUS packet Code field

The number in the packet’s Code field indicates the type of message that has been sent. Clients only send RADIUS Access-Request messages or RADIUS Accounting-Request messages. Servers send three types of RADIUS reply messages or an Accounting-Response message. Table 5-3 is a list of the codes and the message types they identify.

Table 5-3. RADIUS packet Codes and corresponding messages.

Code	**Message**
1	Access-Request (from client)

Table 5-3. RADIUS packet Codes and corresponding messages.

Code	Message
2	Access-Accept (from server)
3	Access-Reject (from server)
4	Accounting-Request (from client)
5	Accounting-Response (from server)
11	Access-Challenge (from server)

The NAS/server exchange described in “RADIUS packets” on page 5-9 illustrates when each type of RADIUS message may be sent.

The NAS sends Accounting-Request when it has information it would like the server to store and the server sends an Accounting-Response to acknowledge receipt of the request.

If the NAS receives no response from the RADIUS server or RADIUS Accounting server, it re-sends its Access-Request or Accounting-Request. The length of time the NAS waits between attempts to re-send, and the maximum number of attempts, may be configured in the NAS. The NAS may also be configured to have primary and secondary authentication and accounting servers.

Identifier

The Access-Request identifier field contains a value which is copied into the server’s response so that the NAS can correctly associate its requests and the server’s responses when multiple users are being authenticated.

Length

The length field is a simple error-checking device. The server silently drops a packet if it is shorter than the value specified in the Length field. If the packet is longer, the server ignores the octets beyond the value of the Length field.

Authenticator

The authenticator field contains a value for a Request Authenticator or a Response Authenticator, depending on the type of message being sent. The Request Authenticator is included in a NAS Access-Request. It is an unpredictable and unique value that NavisRadius adds to the secret key before running the combination through a one-way MD5 hash algorithm. The result supplements the user's password to protect against authentication attacks.

A Response Authenticator is part of Access-Reject, Access-Accept and Access-Challenge messages sent by the RADIUS server. Its value is the result of a one-way hash of the packet's contents and the secret key.

Attributes

Attribute values are also related to the type of message being sent. The number of attribute/value pairs included in the packet's Attribute field is variable, depending on what is required or optional for the type of service requested. There is no minimum requirement for attributes in a message sent by the server to the NAS.

Understanding the dictionary file

The NavisRadius `dictionary` file is a list of the authentication, authorization and accounting attributes you can enter in a `users` file. NavisRadius's installation script installs the `dictionary` file and a companion file named `vendors`. Each `dictionary` entry translates an attribute's human-readable name into an enumerated equivalent and lists all the values for that attribute. NavisRadius uses the number associated with the attribute to parse incoming requests and generate responses. RADIUS protocol requires that the attribute's number be in the range of 1 to 255.

For example, the Framed-Protocol attribute's entry in the `dictionary` file translates it into the number 7 and lists the attribute's nine possible values. For discussion of the `dictionary` file's NavisRadius attributes see Appendix C, "Attributes Reference," and the NavisRadius Manager's HTML help files.

Vendor Specific attributes

NavisRadius follows the RADIUS protocol's method of supporting additional attributes. NavisRadius stores attributes developed by vendors such as Ascend in the `dictionary` file as values of number 26, the Vendor-Specific attribute.

NavisRadius also supports attributes vendors assigned non-standard numbers, such as Ascend-Data-Filter, which the vendor added to the `dictionary` as attribute number 242. NavisRadius maps Ascend-Data-Filter to a value of the RADIUS standard's Vendor-Specific attribute via information in NavisRadius `vendors` file.

Caution: Reply-item attributes that vendors develop for their own NAS machines, such as Ascend-Data-Filter, are not recognized by other vendors' machines. In addition, each NAS must be identified by its vendor name or NavisRadius will not send it a vendor-specific reply-item attribute. See "Understanding the clients file" on page 5-24.

dictionary format

Each translation consists of attribute/value pairs. Each attribute entry and each value entry has four fields. Following is an example of a `dictionary` entry.

```
Attribute attribute-name  integer-encoding type
Attribute Framed-Protocol 7             integer
Value     attribute-name  value-name integer-encoding
Value     Framed-Protocol PPP       1
```

An attribute value is expressed in one of four data types, as shown in Table 5-4:

Table 5-4. The four data types of attribute values.

Type	Description
string	0-253 octets
ipaddr	4 octets in network byte order

Table 5-4. The four data types of attribute values.

Type	Description
integer	32 bit framing in big endian order (high byte first)
date	32 bit value in big endian order (seconds since 00:00:00 GMT — Jan. 1, 1970)
abinary	0-254 octets

Understanding the authfile

NavisRadius provides a means for grouping users together for authentication. The groups are called realms. An `authfile` is a list of the realms to which you can assign users. Members of realms may enter their user names in the format *user@realm* or *realm/user*. However, the separator in a realm-affiliated user's name does not have to be the @ or / character. You may choose different separators with the `radiusd` options `-r` and `-rr`.

Locating authfile

By default, `authfile` and the other NavisRadius data files reside in the server's `/etc/raddb` directory. Although you may change the data directory with the `radiusd -d` option. Appendix A, "Ascend NavisRadius files and commands," has more information about the RADIUS daemon and its options.

You create the `authfile` file yourself with a text editor, or base it on the file named `authfile.ex` which is distributed with NavisRadius. By default, NavisRadius searches for `authfile` when a user profile includes the attribute/value pair, `Authentication-Type = Realm`.

You can create an association between a NAS client and a specific `authfile` named *`prefix`*`.authfile.` The file's name is derived from the Prefix field's value in the NAS's `clients` file entry. If an Access-Request from the associated NAS leads to a user profile containing `Authentication-Type = Realm`, NavisRadius searches for the realm in `prefix.authfile`. If

NavisRadius does not find `prefix.authfile,` it looks for the realm in `authfile.`

Caution: Ascend recommends caution if you want to associate a NAS and a `prefix.authfile` file. If you do not include `prefix.authfile`'s list of realms in the `authfile's` list, NavisRadius may not be able find the realm for a NAS that is not associated with `prefix.authfile.`

The steps for creating a NAS-prefix.authfile link are discussed in “Creating Example Client, User and Realm entries” on page 6-1. You may use a similar process to link a `prefix.users` file and a realm. That link is created when you enter a value in the Prefix field of a realm entry in `authfile` (or `prefix.authfile`!).

Realms

NavisRadius supports many authentication methods, including Kerberos, TACACS, and electronic token keys. When you group users into realms you can select one of the supported authentication methods for all members of that realm. You can also limit editing and viewing privileges by realm to increase security for user profiles. A realm may include all the users in an entire company, or the users in a single corporate department.

For example, suppose a large company provides NavisRadius authentication for three smaller companies. Each of the smaller companies may be identified by an individual realm name in the server’s `authfile` and each of these realms may use a different authentication method.

When one of the company A’s users dials in with the username *jsmith@Aco,* the NavisRadius server finds the Aco realm in its `authfile`. The server then handles the authentication in the manner described in the `authfile`’s Aco realm entry. The entry might tell the NavisRadius server to send the user’s information to another server or to search a local `users` file containing Aco’s user profiles.

authfile format

The `authfile` contains a line of information for each realm. Each line has several white-space delimited fields. Comment lines begin with a leading pound sign (#) and are ignored, as are blank lines.

A line must begin with an entry for the Realm-Name field and include an entry for the Type field. It may also include other field entries as shown in the example below. Some entries in the Type field require that you enter a realm name, DNS hostname or IP address, or file name for the line's Realm/DNS/File field.

```
Realm-name [(alias[,alias])]  [-Protocol] Type [Realm/
DNS/File]
wolves school -pw AFS-KRB umich
buckeye ohio file buckeye
```

Table 5-5. Authfile fields

Field

Description

Value

Realm-Name

Description: An ASCII sting that is a symbol or name for a realm

Usage: DEFAULT indicates how Ascend NavisRadius handles requests for realms not listed in `authfile`.

NULL realm-name indicates how Ascend NavisRadius handles user names that are not in the *username@realm* format.

Wild card syntax **.realm* specifies several related realms.

Example: `*.town.com` stands for `abc.town.com` through `xyz.town.com.`

The format allows one entry to match any of the realms. Place **.realm* entries near end of `authfile` list.

Alias

Description: Optional field of comma-separated realm names within parentheses. Each alias name is equivalent to the realm name and is provided for user convenience.

Example: For realm named California:

(Calif,CA,CAL)

Example: User names that match the realm California:

jsmith@calif, jsmith/ca, jsmith@ca

–Protocol

Description: Forces the processing order of otherwise identical entries

Usage: –PW, –CHAP, –DFLT

Type

Description: The method the Ascend NavisRadius server uses to authenticate users within a given realm.

Usage: All entries are case insensitive.

- passwd - same as unix-pw
- unix-pw - authentication using the UNIX password file

Following types require an entry in the Realm/DNS/File field:

- RADIUS — Authentication by another Ascend NavisRadius or RADIUS server. The Realm/DNS/File field must contain default RADIUS server DNS name. Corresponds to name in `clients` file.
- MIT-KRB — Authentication via MIT Kerberos. The Realm/DNS/File field must contain default Kerberos realm name.
- AFS-KRB — Authentication via AFS Kerberos protocol. The Realm/DNS/File field must contain default Kerberos realm name.

- File — Local Ascend NavisRadius server consults local `prefix.users` file where `prefix` is derived from the entry in the Realm/DNS/File field.
- TACACS —- Authentication at TACACS server via encrypted request. Realm/DNS/File field must contain default TACACS server DNS name.

The method the Ascend NavisRadius server uses to authenticate users within a given realm.

- TACPLUS — Authentication at TACACS+ server via encrypted request. Realm/DNS/File field must contain default TACACS+ server DNS name.
- ACE — Authentication with user's SecureID card
- DEFENDER — Authentication with user's SecureNet key card
- S/Key — Authentication via one-time password generated by the S/Key program
- WinNT— Authentication with Windows/NT domain

Realm/ DNS/File

Description: Entries indicate the realm, hostname, or filename appropriate to the entry in the Type field.

Usage: ASCII text string or IP address in dotted quad notation.

The entry in the Type field determines which of three kinds of Realm/DNS/File field entries is appropriate.

- *Realm* is appropriate when one of the two types of Kerberos authentication is entered in the Type field. *Realm* is the name of the Kerberos realm.
- *DNS* is appropriate when the Type field entry is radius and authentication is performed by a remote Ascend NavisRadius server. *DNS* is the hostname or IP address of the remote Ascend NavisRadius server.
- *File* is appropriate when the Type field is RADIUS and user authentication is done by the Ascend NavisRadius server. *File* is the *prefix* of the `users` file, expressed in the format *`prefix`*`.users.`

Understanding users files

A `users` file is a list of user profiles containing attribute/value pairs. The people the profiles define are users NavisRadius can authenticate for a NAS or server listed in the `clients` file. By default, NavisRadius searches for a file named `users` when it needs to locate user profiles. You create the file and populate it with user profiles by entering the profiles with a text editor, by copying examples from the provided `user.ex` file, or by using one of the NavisRadius graphic interfaces.

You may create more than one `users` files for the NavisRadius server. Additional `users` files usually bear a name in the format *`prefix`*`.users.` The prefix is derived from the value of the Realm/DNS/File field in the `authfile` because, usually, `prefix.users` files are associated with realms. For example, a `users` file associated with a realm called Ournet might have the name *`ournet.users.`*

If you organize users into realms as explained in “Understanding the authfile” on page 5-15, you might want to create a separate `users` files for each realm.

users file attributes

You can only enter in a `users` file the attributes that appear in the `dictionary` file. User profile attribute/value pairs are classified two ways:

- Check-Items — The attribute/value pairs NavisRadius compares to the attributes/values it receives from a client to authenticate a user.
- Reply Items — The attribute/value pairs NavisRadius sends the NAS to authorize a link and services if authentication is successful.

  If authentication fails, NavisRadius might send an attribute/value containing a message about the authentication failure.

users file format

Create the `users` file in the NavisRadius data directory. By default, the data directory is `/etc/raddb,` but you can install NavisRadius in another location. For more information about changing NavisRadius defaults see “Default Ascend NavisRadius settings” on page 2-3.

A users file may contain general comments about the file and separate comments about individual user's profiles. Lines containing comments begin with a pound sign #. Each user profile contains one or more lines. The first line contains authentication attribute/value pairs and can wrap if the entries cannot fit on a single line. Lines below the authentication attribute/value pairs contain authorization attribute/value pairs and must begin with whitespace. All lines between the first and last lines end with commas. The first and last lines of a user profile do not end with a comma. Table 5-6 lists the elements of a user profile. The format is as follows:

```
 #Comment
users-name check-item [, <check-item>]...
   reply-item,
   reply-item
```

Example users file entry

```
#This user began it all.

gwash@whitehse Password=paterusa,

     Service-Type=Framed,
     Framed-Protocol=PPP,
     Framed-IP-Address=105.23.0.1,
     Framed-IP-Netmask=255.255.0
```

Table 5-6. Elements of a user profile.

Element	Description	Value
Comments	Description or notation	ASCII string
• File comment		
• Profile comment		
Lines		
• First line	Initial line of user profile	Check-Item attribute/value pair
• New line	Line(s) between first and final lines; end(s) with a comma	Reply-Item attribute/value pair
• Final line	Last line of user profile	Reply-Item attribute/value pair

Table 5-6. Elements of a user profile.

Element	Description	Value
Check-Item	Entered in first line of User profile Attribute/value pair compared by Ascend NavisRadius to pair sent by client in an Access-Request packet	Any attribute listed in the `dictionary` file can be used as a Check-Item. Some common Check-Item attributes are: • User-Name (required) • Authentication-Type • Password • Service-Type • Ascend-Send-Password • Ascend-Send-Secret • Calling-Station-ID • CHAP Password • Class • Client-Port-NAS • NAS-Identifier • NAS-Port
Reply-Item	User profile newline or final line entry User file attribute/value pair sent by Ascend NavisRadius to client in an Access-Response packet	May include any attribute which is not a check-item and its value

Understanding the clients file

The `clients` file is a list of all the Network Access Servers (NAS) and NavisRadius servers that can send Access-Requests to the NavisRadius server. You create the `clients` file. It is not created or copied to the server during the installation of NavisRadius. Each entry in the `clients` file's list may contain values for five fields, although only values for the Client-Name and Key fields are required. The five `clients` file fields are:

- `System-Name:[port]`
- `Key`
- `Type`
- `Version`
- `Prefix`

clients file format

By default, the `clients` file resides in the `/etc/raddb` directory. It has a simple format. Each entry is one line that consists of field values separated by whitespace that can by a tab or keyboard space. An example entry follows, with the field name shown above each field:

```
System-Name:[port] Key TypeVersion Prefix

marlowe loringl type=nas V2 mystery
```

NAS types and vendor-specific attributes

If you put vendor-specific reply-item attributes in user profiles, you must identify the NAS's vendor with the *`vendor`***`:nas`** option. NavisRadius will not send vendor-specific reply-items to a NAS identified generically by the Type field's **`nas`** option.You can enter one of these values for the Type field:

- `nas`
- *`vendor`*`:`**`nas`**
- `proxy`

For example, Ascend-Idle-Limit is a vendor specific reply-item in Jan Power's user profile. The attribute's value specifies the number of seconds an Ascend MAX, MAX TNT waits before clearing Jan's inactive sessions. Jan's Ascend-Idle-Limit is 30 seconds and she can request connections via two Ascend MAX, TNT units named A and B. NavisRadius's `clients` file contains these entries for the NAS clients:

```
A   inside nas
B   outside ascend:nas
```

NavisRadius sends the MAX, MAX TNT named B the attribute and its value after authenticating Jan Powers. The server does not send the MAX, MAX TNT named A the attribute or value after authentication because A is identified by the generic type option. Therefore, A never clears Jan Power's sessions because it never receives Ascend-specific attributes.

Configuring the clients file

Table 5-7 lists all the clients file fields and their possible values.

Table 5-7. Description of the fields in the `clients` file.

Field	Description	Possible Values
System-Name[:port]	Name of NAS which can send Access-Requests to a Ascend NavisRadius server. Port number option overrides radiusd -pp and -qq options specifying Ascend NavisRadius relay port and accounting relay port. `name1/name2` option allows the same `clients` file to be used by, and distributed to, different Ascend NavisRadius servers.	• IP address in dotted quad notation or valid DNS hostname. • Optional port number format: *name:n* where *n* = UDP or TCP port • Optional two-name entry format: *`name1/name2`* `name1/name2` option valid only if one name matches authentication request's source IP address and other name matches response to hostname command on destination server. `name1/name2` format precludes using the port number format.
Key	Secret key known by a NAS and a Ascend NavisRadius server or by two Ascend NavisRadius servers.	ASCII text string. Minimum of sixteen characters, maximum of 128 characters.
Type	Vendor name and/or type of Ascend NavisRadius machine sending requests to the server.	Format is `type=`*`x`* where *`x`* is `ascend:nas, nas,` or `proxy` If client type and vendor name are unspecified, the server will not send vendor-specific Reply-Items from user profile.

Field	Description	Possible Values
Version	Radius version number.	Format is V*n* where *n* is 1 or 2 The default is 1.
Prefix	Associates the NAS with a user realm. This prefix makes it possible for different NASes to use the same Ascend NavisRadius server, but with access to different databases on the server.	Text string prefix

NavisRadius authentication methods

NavisRadius supports authentication via Password Authentication Protocol (PAP) and Challenge Authentication Handshake Protocol CHAP, as described in the original RADIUS protocol. NavisRadius also authenticates via one-time passwords, token cards, and proxy servers.

PAP and CHAP

Password Authentication Protocol (PAP) is a two-way handshake, ID-and-password protocol, controlled by the user attempting to connect to the network. Under the PAP protocol, the user's machine may repeatedly send an ID-and-password pair via a Point-to-Point Protocol (PPP) connection until the user is authenticated or rejected. The ID/password pair is unencrypted during transmission and the user is only authenticated once at the beginning of the session.

CHAP is more secure than PAP because CHAP requires a three way handshake and authentication is controlled by the NavisRadius server, which may randomly issue challenges to the user during the session. During the three way handshake the user sends his identification to server, receives a challenge from the server, and sends back an answer to the challenge. The user and NavisRadius share a secret key with which the NAS and the server perform a one way hash of the NavisRadius challenge. When NavisRadius receives the user's hashed challenge response, it compares it to the hashed challenge it is expecting. If the expected and received values match, NavisRadius authenticates the user's connection to the server. The server may send additional challenges during the session, which the user must answer to remain authenticated and connected. Challenges and responses are encrypted during transmission.

You implement PAP or CHAP authentication by entering values for two NavisRadius attributes in a user's profile in the `users` file. You must enter a value for the User-Name attribute for PAP or CHAP authentication. PAP also requires a User-Password attribute value and CHAP requires a value for the CHAP-Password attribute.

For complete definitions of the PAP and CHAP authentication methods, see Internet Engineering Task Force (IETF) RFC 1334.

S/Key

NavisRadius supports S/Key, a method of authentication in which the user provides a new password each time he requests access through a NAS. NavisRadius and the user share a list of passwords that must be used sequentially, so that each side knows the current password. S/Key uses a one-way hash function to hide the user's password during transmission. S/Key was developed by Bell Communication Research, Inc.

S/Key's *key* operation produces the list of one-time passwords you distribute to the user and NavisRadius. Both NavisRadius and the user invalidate a password after it has been used, so that it cannot authenticate the user twice.

Be careful not to store S/Key one-time passwords in a manner that allows someone other than the user to see them.

Implementing support for S/Key

- The S/Key library linked to `radiusd` uses a file named `skeypolicy` to determine how to find the S/Key database of user passwords. `skeypolicy` is in the `/etc` directory.
- To set up a user profile for S/Key authentication, make the Authentication-Type attribute a check-item and enter `S/Key` as the attribute's value.

Proxy Ascend NavisRadius servers

NavisRadius supports proxy RADIUS authentication. Proxy NavisRadius servers enable you to authenticate a user whose profile is in a remote NavisRadius server's `users` file. In a proxy arrangement, the NavisRadius proxy server transparently routes NAS Access Request packets to the remote server which authenticates the user. The proxy server then sends the remote server's responses to the NAS. The authenticating server may be another NavisRadius server or a RADIUS server. When NavisRadius supports authentication with a token key, the NavisRadius server is not acting, strictly speaking, as a proxy. It may best be described as an agent, client or translator of an ACE or Defender server that supports the token card process. For more information, see "Token Keys" on page 5-32.

Configuring proxy authentication

In a proxy authentication arrangement, NavisRadius must be informed that a remote RADIUS server will authenticate the user. You must provide NavisRadius with values that define the remote authentication method and explain the location of the remote server. The user's membership in a realm determines where you must enter these values. For more information, see See "Understanding the authfile" on page 5-15.

If the user is identified by realm:

1 Enter **`RADIUS`** as the value for `authfile`'s Type field.

2 Enter the remote server's location as a Domain Name Service hostname in `authfile`'s Realm/DNS/File field.

Following is an example of a realm's `authfile` entry. *RADIUS* indicates a remote NavisRadius server will authenticate the realm's users and *copper.edu* is the server's location:

```
lincoln.com RADIUS copper.edu
```

If the user is not a member of a realm:

1 `In` the user's profile, enter **`RADIUS`** for the Authentication-Type attribute.

2 Enter a value for the default RADIUS server's location in the DEFAULT-RADIUS-SERVER

Other authentication methods supported by NavisRadius

TACACS and TACACS+

TACACS and TACACS+ are versions of an authentication protocol developed by Cisco Systems, Inc. TACACS is a protocol for controlling dial up access through a single, centralized database. RFC 1492 explains that TACACS is a simple client-server protocol that authenticates by comparing received user names and passwords to stored user names and passwords. TACACS is similar to, but less robust than its extension, TACACS+, and NavisRadius.

The TACACS+ protocol provides more RADIUS-like features than TACACS. These features include authentication, authorization and accounting. TACACS+ also supports TCP transport instead of UDP transport, password encryption, and third party token cards. For more information, see “Understanding authorization” on page 5-7 and “Token Keys” on page 5-32.

Following examples of `users` file and `authfile` file entries define a user and the TACACS server that authenticates him.

users file example

The example’s `Authentication-Type=TACACS` check-item specifies that the user is authenticated by TACACS.

```
Paul Reynolds Authentication-Type=TACACS
   Password=philo,
   Framed-Protocol=PPP,
   ...
```

authfile file example

The example identifies a default TACACS server NavisRadius consults whenever an authfile entry contains the Type field value `TACACS`. The Realm/ DNS/File field value indicates an NavisRadius client named Default_TACACS_Server authenticates all of the users in Paul Reynold’s realm.

```
tacrealm TACACS Default_TACACS_Server
```

Note: Review your TACACS or TACACS+ documentation before sending authentication requests from NavisRadius to a TACACS or TACACS+ server.

Kerberos

A Kerberos authentication server is a trusted third party that enables users and servers to identify each other and encrypt the messages they send over a connection. The Kerberos server can identify the user and the server because it maintains a centralized database of users and servers and their secret keys. When it can identify both sides of a requested connection, the Kerberos authentication server supplies each side with a shared secret key for their session. The two sides use the secret key to encrypt and timestamp their transmissions, ensuring the integrity of their exchanged messages.

NavisRadius supports two incompatible versions of the Kerberos 4 authentication protocol, AFS Kerberos and MIT Kerberos. AFS-KRB and MIT-KRB, respectively, are the arguments you enter in the `authfile` *Type* field. AFS Kerberos is the authentication protocol for AFS, a distributed file system originally known as the *Andrew File System*. AFS is the basis for Open Software Foundation's Distributed File System, DFS, and is now marketed and maintained by Transarc Corporation. MIT Kerberos is the version of Kerberos developed at Massachusetts Institute of Technology.

Review your AFS or MIT Kerberos documentation if you want either version to authenticate users. The documentation should describe a configuration file named `/etc/krb.conf.` Make sure the `/etc/krb.conf` file contains valid entries for the Kerberos realms. AFS Kerberos cells are the equivalent of MIT Kerberos realms.

Configuring NavisRadius for user authentication by a Kerberos server is similar to configuring it for TACACS or TACACS+ authentication. Enter the Kerberos realm name in `authfile`'s Realm/DNS/File field instead of entering a DNS hostname for the remote TACACS or TACACS+ server.

Token Keys

NavisRadius supports these vendor’s security token products:

- Axent Technologies Defender Security Server
- Security Dynamics ACE server

Like S/Key, security token keys take the CHAP challenge/response method a step further. Token cards are portable devices which usually contain a secret key and perform a computation that allows the user to respond to an authentication server’s challenge. Token keys may be referred to as Two-Factor Identification, meaning there is something the user has (the card) and something the user knows (a PIN number).

Token keys often resemble credit cards or calculators with small entry keypads. They store information and computational elements in flash memory. In addition to requiring a Personal Identification Number (PIN), most token cards also have internal software that detects attempts to read or change information stored on the card.

Some security token methods also include a timing element. Security Dynamics’ SecureID card and its companion Ace authentication server have matched internal clocks. The token’s response is only valid for a short time, after which it is replaced by a new response.

Axent Technologies DSS Server and SecureNet Key

NavisRadius may be an agent for users who authenticate with AssureNet Pathways’ Defender Security Server and SecureNet Key tokens. SecureNet keys use an encryption algorithm to provide users with a one time password that is the response for the Defender Security Server’s challenge. The one time passwords are created in real time when the user requests access. For security, SecureNet Keys might require a user PIN.

In addition to authenticating users, Defender Security Servers provide user audits, accounting for billing services, and printed reports.

NavisRadius supports multi-threaded functionality for AssureNet Pathways Defender Security Server (DSS), meaning that multiple authentication requests may be in progress simultaneously. Single-thread processing requires that the current request be handled to completion before a new request is accepted.

To support Defender and SecureNet Keys you have place a file named `agent.cf` in the `/etc/raddb` directory. The file is created when you install your Defender server. The Defender server's documentation includes instructions for creating and configuring `agent.cf` and placing it on the NavisRadius server. The `agent.cf` file is akin to the NavisRadius clients file, containing information about the NavisRadius and Defender servers that enable them to recognize each other and work together.

Five different variables may be defined in the `agent.cf` file:

Table 5-8. AssureNet Defender agent.cf file entries.

Name	Type	Description
`agentkey`	key	The key used between the agent and DSS
`agentid`	string	The agent identifier
`dss_address`	string	The hostname or IP address of the DSS
`dss_port`	integer	The port number used by the DSS
`dss_timeout`	integer	The timeout in seconds to wait for a DSS response

Following is an example of an `agent.cf` file:

```
agentkey = 0x01, 0x23, 0x45, 0x67, 0x89, 0xab,
0xcd, 0xef
agentid = foobar
dss_address = 10.11.12.13
```

Security Dynamics SecureID and ACE server

NavisRadius can be a client to Security Dynamic's NavisRadius Encryption (ACE) servers. ACE/Server is a software program and ACE servers authenticate users who carry Security Dynamics' SecureID card. The carrier of a SecureID card uses a PIN and a password for authentication. The authentication procedure does not include challenge/response exchange between the user and the server.

SecureID card holders do not enter a PIN, or user name, and wait for a challenge from the ACE server. They enter a PIN and password together when requesting access. Through programming in its own memory, the SecureID card provides a different time-limited code, or password, every 30 to 60 seconds. Internal matching clocks in the ACE server and the SecureID card are incorporated into the Security Dynamics hash algorithm so that each entity knows the appropriate password at any given moment.

When the PIN and password code are entered together they form a whole authentication key. The key cannot be duplicated by someone who finds or steals a SecureID card because the PIN component will be missing. If someone steals a user's PIN and SecureID code as they are entered, they quickly become useless because the password code changes.

In addition to authenticating users, ACE servers provide accounting utilities and may monitor login and administrative activities.

ACE servers interact with ACE/Clients. These two entities resemble a NavisRadius server and a NAS because the ACE/Client is the user's interface and transparent agent for accessing a network. The ACE/Client initiates, transmits, and manages user authentication requests and the ACE server's responses. Requests and responses are encrypted.

NavisRadius supports multi-threaded functionality for ACE servers, meaning that multiple authentication requests may be in progress simultaneously. Single-thread processing requires that the current request be handled to completion before a new request is accepted.

NavisRadius supports New PIN mode which allows users to change their PIN. NavisRadius also supports the next token mode.

Creating Example Client, User and Realm entries

6

This chapter explains a series of examples which provide the instructions for configuring data files. The examples start with a simple configuration for a Network Access Server, two users, and an Ascend NavisRadius server. Gradually, the examples add more clients and users, and introduce the concepts of realms and proxy servers.

How to use this chapter

This chapter explains how to create entries in Ascend NavisRadius's `clients,` `users,` and `authfile` files. Follow the illustrations to learn which steps you should apply in building the files that provide NavisRadius authentication for your own user community. You should also read this chapter to see the relationship between NavisRadius components and the files that allow the components to connect and communicate

This chapter also describes three companion utilities and a feature of the RADIUS daemon. Two utilities, `radcheck` and `radpwtst`, test whether an NavisRadius server is operating correctly and whether it can recognize a specific user profile's Password attribute value. The third utility, `convert.pl`, converts user profiles created for RADIUS. The daemon feature is debugging, a troubleshooting tool which captures system commands as they are executed and stores them in a file called `radius.debug`.

The examples begin with a simple configuration, then add more NavisRadius features, such as support for user realms, proxy authentication, token card devices, and Database Management Systems (DBMS). To understand the user profiles created in the examples, you do not need to know about all the attributes that NavisRadius supports. However, you can refer to Appendix C, "Attributes Reference," for information about any of the attributes.

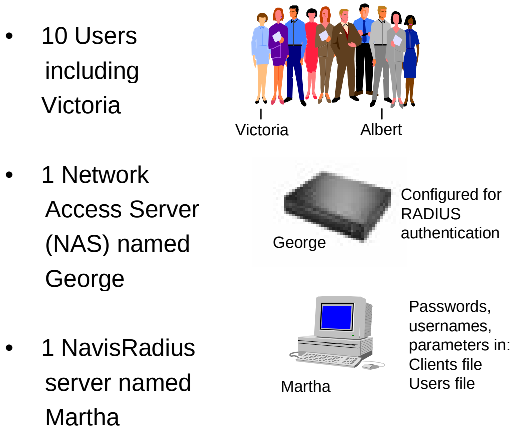
Before you create your own NavisRadius file entries, review gathering NavisRadius information in "Ascend NavisRadius overview" on page 5-2 Then refer to Appendix C, "Attributes Reference,"after collecting information about the users you want to authenticate. That section is a guide to all the attributes NavisRadius supports.

What the examples show

The examples are more than just lists of sample user profiles and `clients` file entries. They include explanations of the entries. For example, they describe such things as the proper format of file entries and the effect of entering a particular value in a file's field. They show how to separate check-items and reply-items in a user profile and how to create new `users` files by entering Prefix field values in the `authfile` file.

Example 1 — a simple model

The first example explains the preparations for authentication of two users, DEFAULT and Victoria. They contact a NAS, named George. The NAS is supported by one NavisRadius server called Martha. Victoria is a member of a small, ten person work group that performs administrative duties. The elements used in this example are displayed in Figure 6-1.



- 10 Users including Victoria
- 1 Network Access Server (NAS) named George
- 1 NavisRadius server named Martha

Victoria
Albert
George
Configured for RADIUS authentication
Martha
Passwords, usernames, parameters in: Clients file Users file

© ASCEND Communications Inc.

Figure 6-1. Users, client and Ascend NavisRadius server in example 1.

Note: configuring the client for authentication

The NAS in this example is represented by a router that supports authentication via NavisRadius. This user guide does not describe configuration of the router or a generic step-by-step guide for configuring a NAS. Many vendors' units can fill the role of the generic NAS in this example. Each type of NAS has its own interface, or suitable method for configuring authentication support, so you should consult the vendor's documentation if you require information about the NAS.

Please remember to configure your own NAS unit(s) to support user authentication by NavisRadius. If you are using an Ascend unit, such as a MAX 4000 or TNT router, authentication configuration is described in the documentation's security supplement.

Example 1 — the steps

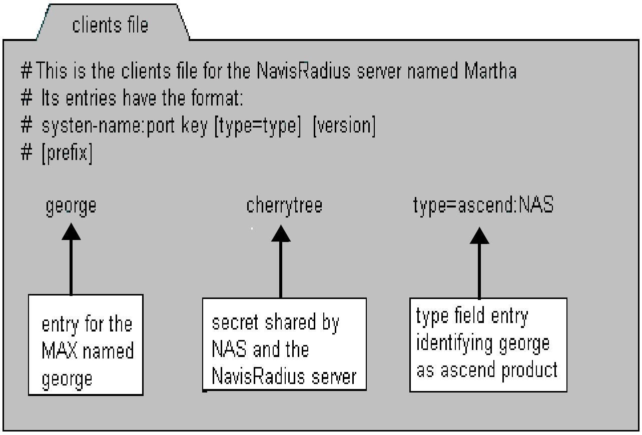
You must perform the following steps to completely configure the NAS and the NavisRadius server before attempting to authenticate a user.

- Configure the NAS according to the instructions in its documentation.
- Edit the `clients` file on the NavisRadius server named Martha, using vi, emacs, or another text editor.
- Edit Martha's `users` file with your text editor.

Example 1's clients file

Figure 6-2 displays the `clients` file entry that creates the client NAS named George. Individual entries in the `clients` file are separated by whitespace. NavisRadius recognizes a tab or space as whitespace. Comments must begin with a pound sign (#).

Note: Sample `clients, authfile` and `users` files are installed at the same time as the NavisRadius software. By default, the installation places these files in the server's `/etc/raddb` directory.



```
clients file
#This is the clients file for the NavisRadius server named Martha
# Its entries have the format:
# systen-name:port key [type=type] [version]
# [prefix]

george          cherrytree          type=ascend:NAS
```

entry for the MAX named george
secret shared by NAS and the NavisRadius server
type field entry identifying george as ascend product

Figure 6-2. `clients` *file entry for the NAS in example 1.*

Understanding example 1's clients file entry

The entry includes three fields of information about the NAS. The *System-name:[port]* field specifies that the name of the NAS is *george*.

You may, in this field, also include the port from which the NAS sends its Access-Requests to the NavisRadius server. If the client, george, should always send Access-Requests from port 5325 enter:

```
george:5325
```

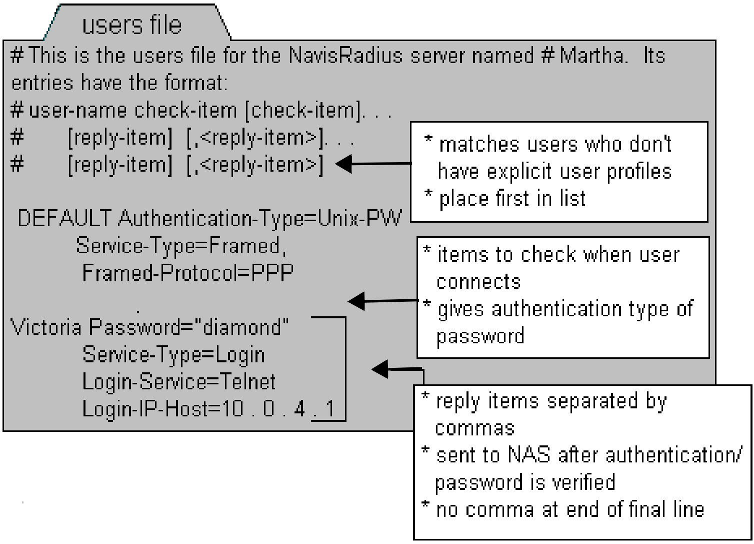
The *Key* field specifies the secret key shared by the NAS and the NavisRadius server. The NAS always sends an encrypted form of the secret key to the NavisRadius server.

The *Type* field has a value of **`ascend:nas`**, which identifies the client's vendor and type. The word **`nas`** must be preceded by a vendor name and a colon if the NAS is to receive vendor-specific attributes in Access-Accept responses. If you only enter **`nas` in the Type field,** the server does not send the client vendor-specific attributes.

You can also enter **proxy** in the Type field, which specifies that the client is another NavisRadius server.

Example 1’s users file

Figure 6-2 displays the NavisRadius `users` file entries that create user profiles for the DEFAULT user and victoria.



users file

```
# This is the users file for the NavisRadius server named # Martha.  Its
entries have the format:
# user-name check-item [check-item]. . .
#       [reply-item]  [,<reply-item>]. . .
#       [reply-item]  [,<reply-item>]

 DEFAULT Authentication-Type=Unix-PW
         Service-Type=Framed,
          Framed-Protocol=PPP

Victoria Password="diamond"
          Service-Type=Login
          Login-Service=Telnet
          Login-IP-Host=10 . 0 . 4 . 1
```

* matches users who don't have explicit user profiles
* place first in list

* items to check when user connects
* gives authentication type of password

* reply items separated by commas
* sent to NAS after authentication/ password is verified
* no comma at end of final line

Figure 6-3. users file entries for DEFAULT and Victoria in example 1.

Victoria’s co-worker’s profiles would be very similar to hers because their activities on the network require the same access to hosts and services. In fact, it is possible to authenticate a small, very similar group of users with only a DEFAULT profile that includes an Authentication-Type attribute with a value such as Unix-PW.

users file format

The initial line of each entry in the `users` file is left-aligned. Subsequent lines in each entry begin with whitespace. NavisRadius recognizes a tab or space as whitespace. You may use either whitespace or commas to indicate the continuation of a series in a list. You may not use them after the last authentication check-item or the last configuration reply-item. See “Components of a users file entry,” below for more information about check- and reply-items. Comments must begin with a pound sign (#).

Components of a users file entry

The user profiles for Victoria and DEFAULT include authentication and configuration information. NavisRadius uses information from the first line of each profile to authenticate Victoria and DEFAULT. NavisRadius sends the information on the subsequent lines in its replies to George, the NAS which uses the information to configure the connection it provides the DEFAULT user or Victoria.

Check-items or reply-items

The attribute/value pairs on the first line of a user profile are called check-items, and the attribute/value pairs on the subsequent lines are called reply-items. Attribute/value pairs are defined in many sections of this user guide. For more information, see the sections “users,” and “dictionary,” in Appendix A, “Ascend NavisRadius files and commands.”

You may enter as many check-items as you wish in a user profile, but they all must appear on the first line. The first line can wrap to a second line if the list of check-items is long. But you cannot artificially cause the line to wrap by entering a line break or carriage return. The first line in the following example is correct, but the second line is not.

```
johnblanker Password=apihfniali, Calling-Station-
ID=6145551212
```

```
johnblanker Password=apihfniali,

Calling-Station-ID=6145551212
```

Check-items are additive. Each condition set with a check-item must be met to authenticate the user. Authentication requirements grow more stringent as you place more check-items in a user’s profile.

All lines that contain reply-items begin with whitespace. You may enter one reply-item per line, or many. If you do place more than one reply-item on a line, you may separate them with whitespace alone or with whitespace and commas. Commas are optional, but recommended for clarity in a long list.

Commas

Commas indicate continuation. You must add a comma at the end of line of reply-items if more reply-items appear the following line.

The last check-item in the first line and or the last reply-item in the user profile must not be followed by a comma. The absence of a comma indicates the end of the check-items and the end of the profile.

User-Name attribute

NavisRadius always interprets the first attribute in a user profile as the User-Name. The attribute is represented only by its value. For example, the DEFAULT user profile begins with the value DEFAULT rather than the attribute/value pair, User-Name = DEFAULT.

The DEFAULT user profile

The DEFAULT entry ensures that everyone can be matched to a profile in the `users` file. For example, there is no individual entry in the `users` file for Jane, one of Victoria’s nine coworkers. When the NAS asks NavisRadius to authenticate Jane, the program searches for an entry that begins with the word Jane, finds none, authenticates Jane based on the DEFAULT check-items, and sends George a response containing the reply-items in the DEFAULT profile.

Located beneath the comments, the DEFAULT entry begins with the User-Name attribute and one check-item attribute, Authentication-Type. Following the first line are the reply-items Service-Type and Framed-Protocol. The User-Name attribute, DEFAULT, identifies the user described by this profile.

The DEFAULT user profile should be at the beginning of the user profiles so it can quickly be found when the user named in an Access-Request does not have a personal profile. NavisRadius searches the entire file for the non-existent profile then returns to the beginning of the file to find a DEFAULT profile. Placing it first reduces NavisRadius's search time.

Authentication-Type Check-item

The check-item `Authentication-Type = Unix-PW` tells NavisRadius to authenticate the user if information in an Access-Request packet from the client matches the password for DEFAULT in the server's `/etc/passwd` file.

Only check-items may appear on the first line of a user profile. NavisRadius uses DEFAULT and the Authentication-Type attribute/value pair to authenticate a user who matches the DEFAULT profile.

The Authentication-Type attribute and the Password attribute discussed in "Victoria's user profile" on page 6-10 are common NavisRadius authentication check-items, but they are rarely used together in the same profile. The Authentication-Type values "Realm" and "ACE" appear in user profiles in examples 2 and 3, which discuss user realms and token card devices.

Service-Type reply-item

The reply-item `Service-Type=Framed` indicates that a caller who matches the DEFAULT profile must use a framed protocol such as Point-to-Point Protocol (PPP), AppleTalk Remote Access (ARA) or Frame Relay (FR).

This attribute /value pair is indented, or preceded by whitespace, so it is a reply-item. NavisRadius will send this information to the client George in an Access-Accept packet. George will use this to configure the caller's connection, preventing the caller from connecting as a login user.

Framed-Protocol reply-item

The `Framed-Protocol=PPP` reply-item indicates that a caller who matches the DEFAULT profile must use the framed Point-to-Point Protocol (PPP).

NavisRadius also sends this information to the client.

Victoria’s user profile

Victoria’s user profile is as simple and uncomplicated as the DEFAULT profile. She must be authenticated so she can Telnet into the client’s terminal server to start a Telnet session on another host with the IP address of 10.0.4.1.

User-Name check-item

The check-item `victoria` is the value for the User-Name attribute that must begin Victoria’s user profile.

Password check-item

This `Password = “diamond”` check-item identifies the string that Victoria must provide as her password.

The password must be surrounded by quotes although Victoria does not need to type them when she is prompted for her password.

You must not enter a quotation mark within the password string when manually editing the `users` file because the second quotation mark identifies the end of the string. For example, if you use vi to change Victoria’s password to “diam”ond” NavisRadius will accept the string “diam”, but reject “diam”ond”.

Service-Type reply-item

The `Service-Type=Login` reply-item indicates that George must create a Login connection for Victoria.

George may not set up a framed connection like the one given the DEFAULT user, who connects via the framed protocol PPP.

Login-Service reply-item

The `Login-Service = Telnet` reply-item allows Victoria to immediately begin an asynchronous Telnet session with the host specified by the Login-Host attribute after she is authenticated.

Login-IP-Host reply-item

The `Login-IP-Host = 10.0.4.1` reply item identifies the host that George connects Victoria with via Telnet after she is authenticated.

Example 2 — building on the simple model

In our second example, Victoria's work group has grown, and a realm called admin has been created to distinguish Victoria's group from another that contains a user named Albert. Another NAS, a router named Tom, has also been added to the network. Like George, Tom is supported by the authentication server called Martha. The elements used in this example are displayed in Figure 6-4.

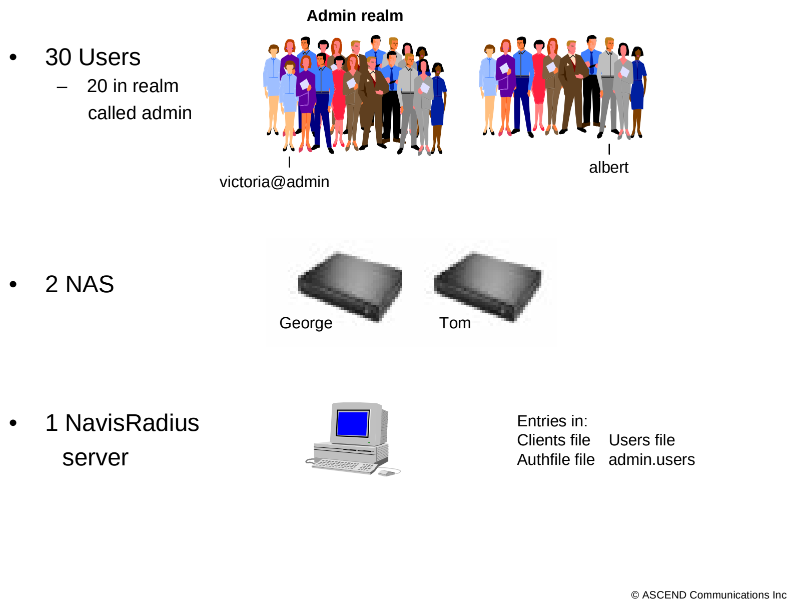


Figure 6-4. Example 2. A Realm called Admin and a second NAS appear.

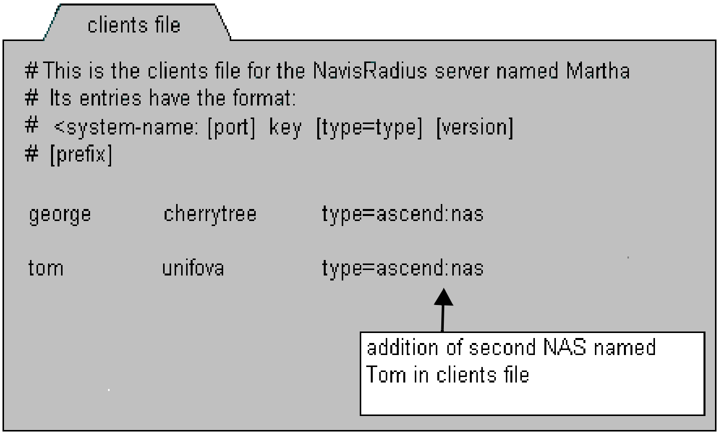
Example 2 — the steps

You must perform the following steps to configure the new NAS, Tom, and the NavisRadius server Martha, before attempting to authenticate.

- Configure the NAS named Tom according to the instructions in its documentation.
- Edit the `clients` file on Martha, the NavisRadius server, using vi, emacs, or another text editor.
- Edit the `authfile` file on Martha with your text editor.
- Edit Martha's `users` file.

Example 2's clients file

Figure 6-5 shows the `clients` file on Martha, the NavisRadius server. It has been edited to add the new client named Tom. Tom and George are identical machines, and their entries are the same. You could add comments to describe each NAS. Comment lines begin with a pound sign (#).



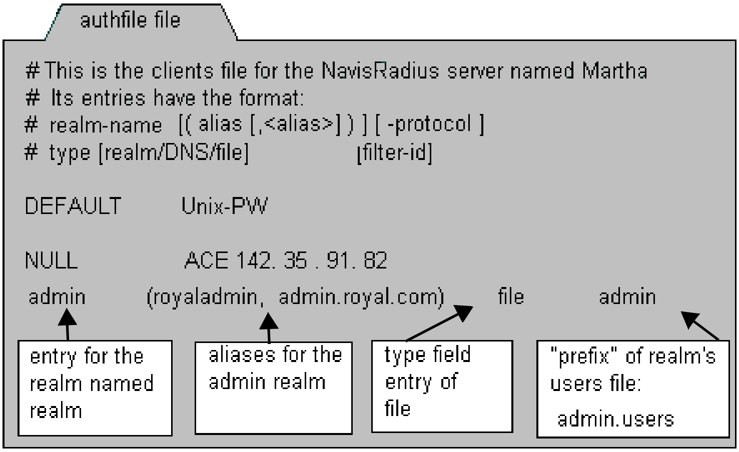
```
# This is the clients file for the NavisRadius server named Martha
#  Its entries have the format:
#   <system-name: [port]  key  [type=type]  [version]
#  [prefix]

george        cherrytree        type=ascend:nas

tom           unifova           type=ascend:nas
```

addition of second NAS named Tom in clients file

Figure 6-5. Example 2 `clients` file, with the entry for the NAS named Tom.

Example 2’s authfile file

An `authfile` is the means by which NavisRadius supports user realms. It is similar to the `clients` file, in that each entry consists of values for fields that, together, describe a single entity. Example 1 did not include a step to create an entry in the NavisRadius `authfile,` because Victoria’s work group was small and there was no need to divide it into realms.


authfile file

```
# This is the clients file for the NavisRadius server named Martha
# Its entries have the format:
# realm-name  [( alias [,<alias>] ) ] [ -protocol ]
# type [realm/DNS/file]              [filter-id]

DEFAULT        Unix-PW

NULL            ACE 142. 35 . 91. 82
 admin       (royaladmin,  admin.royal.com)       file        admin
```

entry for the realm named realm

aliases for the admin realm

type field entry of file

"prefix" of realm's users file: admin.users

Figure 6-6. Example 2 `authfile` entry creating the admin realm and the users file named admin.users, which will contain Victoria’s user profile.

Creating multiple users files

A file named `users` is created when NavisRadius is installed, but NavisRadius also supports additional `users` files you can create after installation. The files are created when you enter `File` in the `authfile`’s Type field and a name in the `authfile`’s Realm/DNS/File field. The Realm/DNS/File entry is added as a prefix to the suffix `users`, creating a new `users` file. In this example, the new file is `admin.users`.

Examining the admin authfile entry

This one-line entry has far-reaching effects. In addition to creating the admin realm and the `admin.users` file, it defines the user names admin realm's users will provide to be authenticated.

- `admin`
  - This entry for the Realm-Name field is the name assigned to the realm.
- `royaladmin, admin.royal.com`
  - This entry for the optional Alias field provides alternatives which can be used interchangeably with the actual realm name, admin. The purpose for allowing aliases is described in the section "Realm user name formats" on page 6-15.
- `File`
  - This entry for the Type field causes NavisRadius to search for a `users` file with a prefix that matches the entry for the Realm/DNS/File field.

  You must enter a value for the Type field, choosing from a long list of options that includes: Passwd, Unix-PW, RADIUS, MIT-KRB, AFS-KRB, File, TACACS, TACPLUS, ACE, DEFENDER, and S/KEY. These options are described in "authfile," in Appendix A, "Ascend NavisRadius files and commands." Most of the options require that the Realm/DNS/File field also contain an entry that identifies the location of a server or a file name. For example:

  - If the realm's Type is RADIUS, NavisRadius acts as a proxy to a RADIUS server. The Realm/DNS/File field must contain a Domain Name Service address for the RADIUS server.
  - If the realm's Type is ACE, members of this realm are authenticated with a SecureID card. Do not enter information in the Realm/DNS/File field. A file named `/var/ace/sdconf.rec` on the NavisRadius server contains all the information about the ACE server.
- `admin`
  - This entry for the authfile Realm/DNS/File field prompts NavisRadius to create a new file called `admin.users`. The file will contain the admin realm's users' profiles.

Realm user name formats

In example 1, the only individual user name in the `users` file's is Victoria. This simple name is enough to identify her profile when she requests her Telnet session. Now she is a member of the admin realm and the `authfile` entry for admin provides an identifier to add to her username.

Victoria can use either of two default formats to match the User-Name attribute in her user profile, `victoria@identifier` or `identifier/victoria`. The @ and the / characters are NavisRadius's default user name separators for members of realms.

The `authfile` entry allows Victoria to choose from among three identifiers that she can use interchangeably. The three identifiers in the admin entry are `admin, royaladmin`, and `admin.royal.com`., so Victoria can enter any of the following user names:

- `victoria@admin, admin/victoria`
- `victoria@royaladmin, royaladmin/victoria`
- `victoria@admin.royal.com, admin.royal.com/victoria`

Other authfile entries

`authfile` should contain at least two special entries, DEFAULT and NULL. DEFAULT is similar to `users` file DEFAULT user profile. If users supply a realm name that is not in `authfile`, they match the DEFAULT `authfile` entry. NULL provides a match for those users who:

1 provide a user name unaffiliated with a realm

2 match a DEFAULT profile with an Authentication-Type value of `Realm`

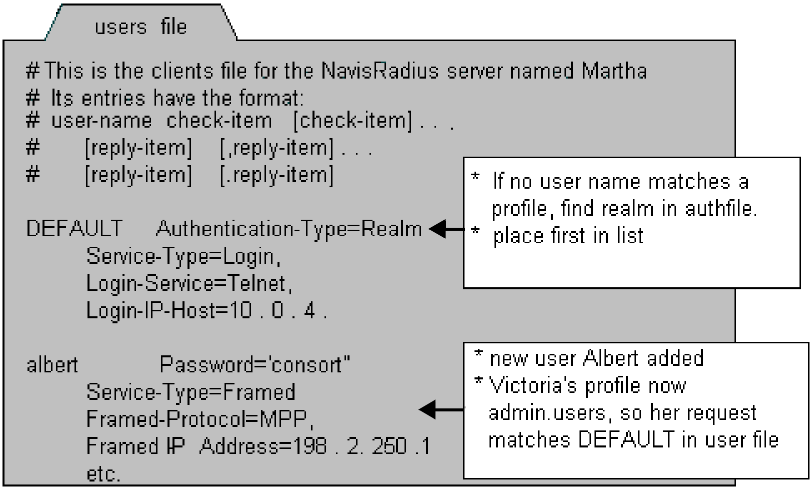
Typically, there are two reasons to use the NULL realm:

1 You can create a minimal `users` file containing only DEFAULT entries to speed up processing of a realm's user profiles.

2 You can ease the migration from an environment with no realms to one that does have realms.

Example 2's users file

Figure 6-7. Example 2's edited `users` *file.*



users file

```
# This is the clients file for the NavisRadius server named Martha
# Its entries have the format:
# user-name  check-item  [check-item] . . .
#      [reply-item]   [,reply-item] . . .
#      [reply-item]   [.reply-item]

DEFAULT    Authentication-Type=Realm
       Service-Type=Login,
       Login-Service=Telnet,
       Login-IP-Host=10 . 0 . 4 .

albert          Password='consort"
       Service-Type=Framed
       Framed-Protocol=MPP,
       Framed IP  Address=198 . 2. 250 .1
       etc.
```

* If no user name matches a profile, find realm in authfile.
* place first in list

* new user Albert added
* Victoria's profile now admin.users, so her request matches DEFAULT in user file

Examining the users file again

Figure 6-7 displays the NavisRadius `users` file following the creation of the realm for Victoria's work group and the addition of Albert's work group. The file has been changed in many ways:

- Victoria's user profile has been deleted from the `users` file and added to the file named `admin.users.`

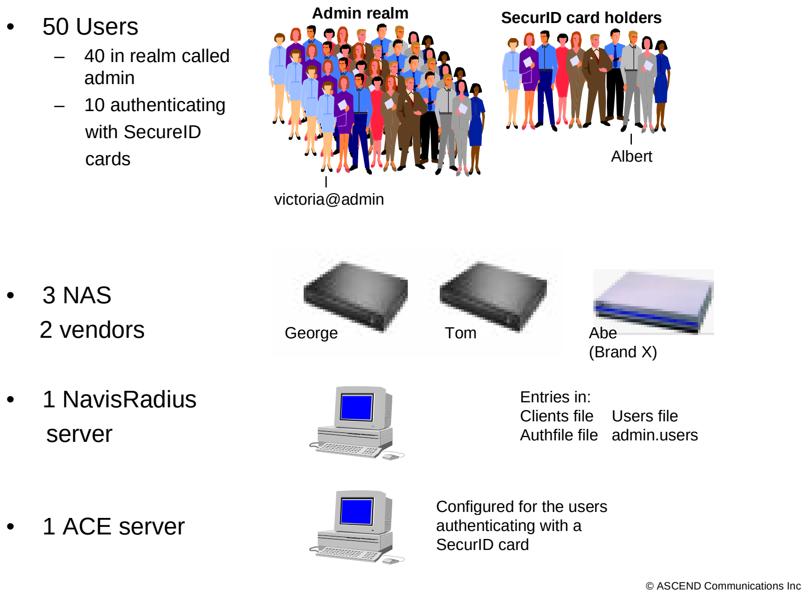
  Her User-Name in that file is not, as you might expect, `victoria@admin`, or any of the other possibilities shown in "Realm user name formats" on page 6-15. NavisRadius strips the realm identifier from her user name before sending it to `admin.users`.

- The DEFAULT profile has also been edited. Its Authentication-Type value is now `Realm`. `authfile` is consulted when NavisRadius cannot find an individual's personal user profile in `users` file. All of the members of Victoria's admin realm match DEFAULT because their profiles are now in `admin.users`.

- Albert's user profile has been added.

Example 3 — expanding the model again

In the third example, Victoria’s work group has grown again. Albert’s group, while not a realm, now authenticates using an token card device called SecureID. A different vendor’s router, named Abe has been added to the network as a client and an ACE server handles SecureID authentication for Albert’s group. The elements used in this example are displayed in Figure 6-8.



- 50 Users
  - 40 in realm called admin
  - 10 authenticating with SecureID cards
- 3 NAS
  2 vendors
- 1 NavisRadius
  server
- 1 ACE server

© ASCEND Communications Inc

Figure 6-8. Example 3: More users, a new clients, a new server, and a new authentication method, electronic keys.

Example 3- the steps

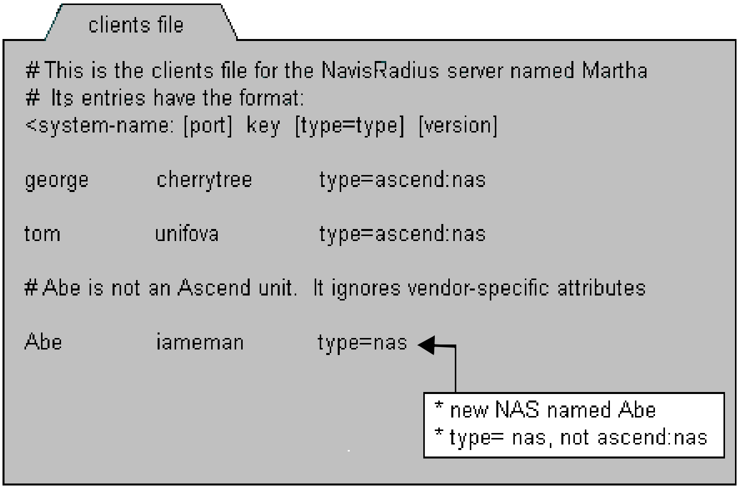
You must perform these steps to configure the new NAS, Abe, and the NavisRadius server Martha, prior to authenticating users.

- Configure the NAS named Abe as instructed by its documentation.
- Edit the `clients` file on the Martha, the NavisRadius server.
- Edit Martha's `users` file.
- Add a binary file named `sdconf.rec` to Martha.

  The file is created on th

Figure 6-9 displays Martha's `clients` file after it has been edited to add Abe. Abe's entry is different from the Ascend units, George and Tom. They understand the `dictionary` file's vendor-specific Ascend attributes. The comment about Abe explains that this NAS only understands attributes from the standard RADIUS protocol. Abe ignores vendor-specific attributes sent to it as reply-items from user profiles.

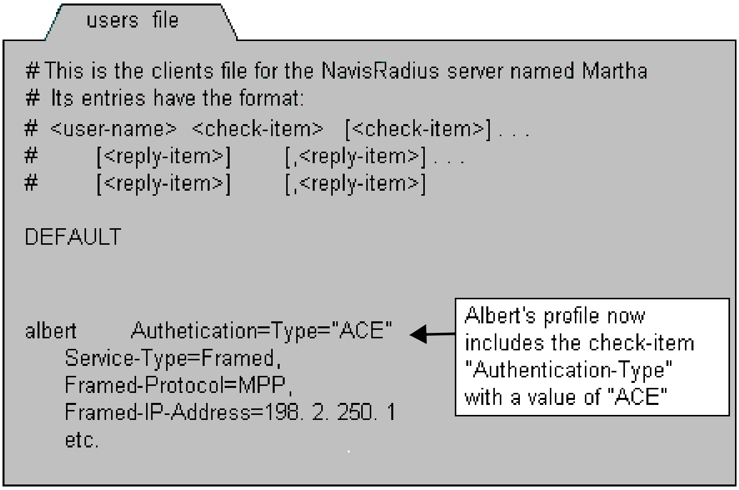
Figure 6-9. Example 3's clients file with the addition of the NAS, Abe.



```
# This is the clients file for the NavisRadius server named Martha
#  Its entries have the format:
<system-name: [port]  key  [type=type]  [version]

george        cherrytree        type=ascend:nas

tom           unifova           type=ascend:nas

# Abe is not an Ascend unit.  It ignores vendor-specific attributes

Abe           iameman           type=nas
```

* new NAS named Abe
* type= nas, not ascend:nas

Example 3’s users file



users file

```
# This is the clients file for the NavisRadius server named Martha
# Its entries have the format:
# <user-name>  <check-item>   [<check-item>] . . .
#       [<reply-item>]       [,<reply-item>] . . .
#       [<reply-item>]       [,<reply-item>]

DEFAULT



albert       Authetication=Type="ACE"
     Service-Type=Framed,
     Framed-Protocol=MPP,
     Framed-IP-Address=198. 2. 250. 1
     etc.
```

Albert's profile now includes the check-item "Authentication-Type" with a value of "ACE"

Figure 6-10. *Example 3’s* `users` *file showing Albert’s new user profile attribute allowing him to authenticate via a SecureID card. (The DEFAULT entry has been truncated for this illustration, but it would contain valid attribute/value pairs.)*

The Authentication-Type value of “ACE” is necessary for anyone who uses a SecureID card to authenticate. NavisRadius also supports devices distributed by AssureNet Pathways. If a user has an AssureNet Pathways device, the appropriate Authentication-Type value is “DEFENDER”, which is the name of the AssureNet server.

A note about Ascend NavisRadius proxy servers

When NavisRadius works in concert with servers like ACE and DEFENDER, it is not really operating as a proxy server. That can only truly be said if NavisRadius is working with a RADIUS server or another NavisRadius server. The pertinent factor is whether or not the other server actually does contain user profiles that supply a NAS with reply-item attributes. ACE, DEFENDER and Kerberos servers play a part in authentication of users when they operate in concert with NavisRadius, but they do not send through the NavisRadius server, the attribute/value pairs the NAS uses to configure the user's connection.

Ascend NavisRadius server's sdconf.rec file

The NavisRadius server's `sdconf.rec` file is not a component of NavisRadius itself. You install the file on the NavisRadius server when you configure an ACE server for users authenticating with SecureID token cards. This file must be added to any server which, like NavisRadius, supports ACE/SecureID user authentication. This user guide is not the proper source of information about configuring the `sdconf.rec` file or the ACE server. Please refer to the documentation supplied by the vendor of that product, namely, Security Dynamics Inc.

Note: The sdconf.rec file has to be placed in the `/var/ace` directory.

Compatibility with Ascend RADIUS

The differences between Ascend's free RADIUS server source code and NavisRadius are explained in the following sections.

Changing Passwords

Ascend RADIUS supports the Ascend-PW-Expiration attribute, but NavisRadius does not. NavisRadius replaces Ascend-PW-Expiration with the Expiration check-item. NavisRadius does support password changing and the associated attributes, Ascend-PW-Warntime and Ascend-PW-Lifetime.

A user can change his password during the NavisRadius authentication process if you run the RADIUS daemon with the `-P` command line option and store the password in the user's profile. NavisRadius does not support password changing if the password in stored in another location, such as the UNIX password file or a supported file.

This daemon command line supports password changing:

```
radiusd -P
```

This user profile allows password changing:

```
fredie Password=“icwtifacgts”, Expiration=“Jun 10 1997”,

    Ascend-PW-Warntime = 5,
    Ascend-PW-Lifetime = 60,
    Framed-Protocol = SLIP
```

Specifying a Token Card

Ascend RADIUS reserves Password attribute values such as “ACE” and “DPI” if you enable the reserve value feature when you compile the RADIUS source code. You can use these reserved values to specify the token card a user carries. Although NavisRadius recognizes these Password values, it does not reserve a special significance for them. Instead, NavisRadius automatically recognizes Authentication-Type attribute values that perform a similar function.

If you switch from RADIUS to NavisRadius, you can use a script called `convert.pl` to convert RADIUS token card users’ Password values to Authentication-Type values. See “AppleTalk Remote Authentication (ARA)” on page 6-22.

Table 6-1 illustrates the change:

Table 6-1. Different means of specifying token cards

Ascend NavisRadius	**Free RADIUS**
Authentication-Type=UNIX-PW	Password=”UNIX-PW”
Authentication-Type=ACE	Password=”ACE”
Authentication-Type=DEFENDER	Password=”DEFENDER”
Ascend NavisRadius does not support SAFEWORD	Password=”SAFEWORD”

AppleTalk Remote Authentication (ARA)

Ascend RADIUS assigns two meanings to this reply-item entry in a user profile:

```
Framed-Protocol=ARA
```

One meaning is obvious; the authenticated user connects via the ARA protocol. But RADIUS also deduces from the attribute’s value that the authentication method for this user is ARA.

NavisRadius separates the ARA protocol and the ARA authentication method. If a user is authenticated by ARA, the user profile must contain two attribute/value entries. Authentication-Type is a check-item that appears on the first line of the profile and protocol is a reply-item that appears below the first line, as this example illustrates:

```
arauser Authentication-Type= ARA-DES, Password=”dsmajbyl”

    Framed-Protocol=Ascend-ARA
    Ascend-Send-Secret= “dsmajbyl”
```

Identifying Network Access Servers (NAS) Vendors

Ascend RADIUS does not provide a means for identifying a NAS vendor. The server sends to the NAS client all the standard and vendor-specific reply-items it discovers in a user profile, even though the vendor-specific attributes might not be understood by the client.

NavisRadius supports an entry in the `clients` file `Type` field that identifies the NAS vendor. When the vendor name appears in the entry, as shown below, NavisRadius only sends the user profiles’ appropriate vendor-specific and standard reply-items to the NAS client.

```
larry tsksbtnas type=Ascend:NAS
```

Converting RADIUS user profiles

NavisRadius supports user profiles files created for the Ascend RADIUS server. The Ascend RADIUS and NavisRadius `dictionary` files are slightly different because NavisRadius conforms with the RADIUS RFC standard, but Ascend

RADIUS source code is based on the standard’s draft documents. The IETF RADIUS RFC standard modifies a few of the attribute names and values that were proposed in the standard’s draft documents. Table 6-2 and Table 6-3 compare NavisRadius and Ascend RADIUS attribute and value names.

Table 6-2. Attribute name changes

Attribute Number	Ascend NavisRadius	Ascend RADIUS
3	CHAP-Password	Challenge-Response
6	Service-Type	User-Service
8	Framed-IP-Address	Framed-Address
9	Framed-IP-Netmask	Framed-Netmask
11	Filter-ID	Framed-Filter
14	Login-IP-Host	Login-Host
20	Callback-ID	Callback-Name
30	Calling-Station-ID	Client-Port-DNIS

Table 6-3. Attribute value name changes

Value	Attribute	Ascend NavisRadius	Ascend RADIUS
1	Service-Type	Login	Login-User
2	Service-Type	Framed	Framed-User
3	Service-Type	Callback-Login	Dialback-Login-User
4	Service-Type	Callback-Framed	Dialback-Framed-User
5	Service-Type	Outbound	Dialout-Framed-User

Table 6-3. Attribute value name changes

Value	Attribute	Ascend NavisRadius	Ascend RADIUS
6	Service-Type	NAS-Prompt	Shell-User
1	Framed-Compression	Van-Jacobson-TCP-IP	Van-Jacobsen-TCP-IP
255	Framed-Protocol	Ascend-ARA	ARA

A perl language script called `convert.pl` automatically converts the Ascend RADIUS attributes and values so they match the standard's documentation.The `convert.pl` script is distributed with NavisRadius and is placed in the server's `/etc/raddb` directory when you install the program.

Note: You can use the `convert.pl` script's `-s` option to change old reserved Password values to reserved Authentication-Type values. Use the option cautiously, though, because the Password values were only reserved if you enabled the feature when building the `radiusd` executable when compiling the source code.

Running convert.pl

You can run `convert.pl` two ways. One method converts the selected `users` file's entries and overwrites the file. The other method creates a backup of the original file before the converting the file's entries.

Install a copy of the perl interpreter on the system before you run the `convert.pl` script. The script should run with PERL v4 or PERL v5.

Without backup

Invoke the `convert.pl` script by typing this on the command line, substituting the users file name as the variable. The lesser than and greater than signs are part of the command and you must enter them The converted file is sent to stdout.

```
convert.pl < users.old > users.new
```

With backup

Run this way, the script copies the `users` file and adds the extension.`bak` to its name. Then it converts the original file's entries and saves the file with the original file's name. If `users.bak` already exists, it is overwritten.

```
convert.pl users
```

Testing Ascend NavisRadius

NavisRadius provides two diagnostic tools called `radcheck` and `radpwtst` for testing an NavisRadius server. Run these diagnostic tools from the workstation's command line after installing NavisRadius and configuring its files. `radcheck` determines if your NavisRadius server, or any other you can access, is operational. `radpwtst` confirms that the server can authenticate a specific user. You can run each command with a variety of options to test a particular NavisRadius configuration. (Both commands are discussed in Appendix A, "Ascend NavisRadius files and commands,")

See Also: `"authfile, clients, dictionary, users, radcheck, radpwtst, radcheck" on page A-30 and` "radpwtst" on page A-36.)

Verifying Ascend NavisRadius is operational with radcheck

The `radcheck` command starts a process on an NavisRadius server to test if another NavisRadius server is operational. The target of the test does not have to be registered in the `clients` file on the server running the test. Type `radcheck` and the target server's Domain Name Service hostname. You can run the command with options that change `radcheck`'s default settings (Table). When debugging and version are enabled, `radcheck` prints the information to the server's standard output.

The command has the following format:

```
radcheck  [ -d] [ -p] [ -r] [ -t] [ -v] [ -x]
          servername
```

`radcheck` options and defaults

Option	Description	Default
`-d`	Ascend NavisRadius directory on server being tested	`/etc/raddb`
`-p`	Ascend NavisRadius port on server being tested	1812
`-r`	Number of verification attempts `radcheck` makes	10
`-t`	Length of time before `radcheck` times out	3 seconds
`-v`	Version of Ascend NavisRadius	Not Enabled
`-x`	Turn on debugging	Not Enabled

radcheck example

The RADIUS daemon on an NavisRadius server named potter is running with the `-d` and `-p` options. The arguments for potter's `-d` and `-p` options are `/etc/aacrd` and `6495`. These arguments indicate that potter's NavisRadius files are in a directory named `/etc/aacrad` and that its authentication port is 6495.

You must `radcheck` with the same arguments as the daemon you want to test, so you might test potter by running the following command on another NavisRadius server.

```
radcheck -d /etc/aacrad -p 6495 potter
```

The example generates the following response:

```
auth queue: 9/7, acct queue: 3/3, maxtime: 0 (Fri Oct 25
16:28:36 1996)

authfile: 5, clients: 10, users: 200, fsmid: Ascend, Fri Oct
25 16:28:36 1996

Version 3.0 NOSHADOW sun

“potter(6495)” is responding
```

In addition to confirming that potter is operational, the response provides the following statistics about the server's status and configuration:

- Number of authentication requests and replies handled
  - 9 and 7 (from auth queue)
- Number of accounting requests and replies handled
  - 3 and 3 (from acct queue)
- Date and time of the test
  - Fri Oct 25 16:28:36 1996
- Number of entries in
  - authfile: 5
  - clients: 10
  - users: 200
- Finite State Machine Identification
  - Ascend
- Version of Ascend NavisRadius on potter
  - Version 3.0
- Operating system on potter
  - Solaris

If the test were unsuccessful, the response would be:

`“potter(6495)”` *`message`*

Message has the following four variations:

1. `No reply from Ascend NavisRadius server “potter(6495)”`

   Although its request has been sent to the server, `radcheck` has timed out waiting for a response.You might try changing the timeout setting.
2. `Received non-matching id in server response`

   The server's reply packet contained an authenticator that did not match the hashed shared secret key that `radcheck` expects to receive from the server. `radcheck` cannot verify that the response is from the server named potter.
3. `Received invalid reply digest from server`

   The server's reply packet contained a digest code that did not match the digest code of the request packet sent by `radcheck`. `radcheck` cannot verify that the server's reply is a response to this specific `radcheck` request.

4. `No such server: “potter(6495)”`

   `radcheck` cannot find an NavisRadius daemon listening at UDP port 6495 on the server named potter or Domain Name Service cannot locate the server named potter.

Verifying user passwords on an NavisRadius server using radpwtst

You can use `radpwtst` to verify any user's password on a server running NavisRadius or RADIUS. `radpwtst` performs the verification on any user name that is entered on the command line as a `radpwtst` option. You can enter the username in the individual or realm format. The server prompts you to enter the user's password after it has located the user's profile, then sends a message confirming or denying the password you entered.

radpwtst options

Other `radpwtst` options allow you to specify where you want to search for the user's password information and where you want the server to send its answer. The options are very similar to those you can use with the `radcheck` command and are generally associated with changing the settings for:

- Server's data directory
- Users file, which might be named `users` or `prefix.users` if the username indicates membership in a realm
- Communications port
- UDP port
- Address of the client expecting the answer
- Number of retries
- Timeout limit

The format of the radpwtst command is:

```
radpwtst  [-c code] [-d directory] [-f file] [-g group]
```

```
[-h] [-i client IP address] [-l async-port]
[-p UDP_port] [-r retries] [-s servername]
[-t timeout] [-u type] [-v [1|2]]
[-w password] [-x] [-:<attribute>=<value>]
username[@realm]
```

radpwtst example

```
radpwtst  -f admin.users -i 173.157.2.11 -r 5
          -t 5 kate
```

The command sets the number of retries to five and the timeout to five seconds. It also tells the client to pass the following message to the server:

Look up the password of a user named Kate in a file named admin. users. Send a prompt for the password to IP address 173.157.2.11.

If authentication of the user password succeeds, the message displayed is

```
authentication OK
```

If the authentication fails the message is

```
“userid” authentication failed
```

Debugging Ascend NavisRadius

You can capture a list of commands the system executes during NavisRadius authentication, authorization and accounting operations by enabling a feature of the RADIUS daemon called debugging. The list of commands is a tool for reviewing system activity and investigating NavisRadius configuration errors. You can also control what, and how much, information the list contains by setting the debugging level. Output from the debugging process is stored in a file you can read with any text editor. The file that stores debugging output is `radius.debug,` which is created when you enable debugging. You cannot change this default name or direct the debugging output to a different file.

By default, debugging is not enabled. You can enable it and set the debugging level by restarting the daemon process or by sending the process a signal

Enabling debugging from the command line

Following is the command line entry for restarting the daemon, enabling debugging, and setting the debugging level to 1:

```
radiusd -x
```

You can type the -x option on the command line more than once. Each additional -x increases the level of debugging, adding to the depth and variety of the information collected in the `radius.debug` file. This entry raises debugging to the third level:

```
radiusd -x -x -x
```

Level 3 debugging captures minimal debugging output, attribute/value pairs sent and received, high level output of the Finite State Machine, and function tracing

Note: You can enter more than one `radiusd` option at a time. The following entry changes the data directory, allows Token-caching and sets debug at level 3:

```
radiusd -d /etc/raddb/data -C -x -x -x
```

Enabling debugging with signaling

You can enable debugging while the daemon is operating by sending `radiusd` a SIGUSR1 signal. This signal affects debugging like the -x command option, increasing the debugging level one step every time you send the signal. You set the debugging level to 3 by sending `radiusd` three SIGUSR1 signals.

Disabling debugging

Disable debugging by restarting `radiusd` without the -x option or set the debugging level to 0 by sending the daemon a SIGUSR2 signal.

Comparison of debug and accounting detail information

`radius.debug` does not capture the same information as the accounting log, which collects data about authenticated connections from the NAS client. You enable RADIUS Accounting during the NavisRadius installation. Following are truncated examples from a `radius.debug` file and an accounting detail file. Debugging is set to level 2.

radius.debug example

```
directory = /etc/raddb Program = /usr/sbin/radiusd

rad_recv: entered

get_radrequest: entered
rad_recv: entered
get_radrequest: entered
rad_recv: entered
get_radrequest: entered
gen_valpairs: entered

    User-Name = “t031982”
    User-Password = “\x16 \xe9u\xf0”
    NAS-IP-Address = 192.0.0.1
    NAS-Port = 10501
    Service-Type = Framed
    Framed-Protocol = PPP
    State = ““
    Called-Station-Id = “5551212”
    Acct-Session-Id = “29600”
Accounting detail example

Sat Feb 15 07:50:46 1997

    User-Name = “t031982”
    NAS-IP-Address = 144.158.43.168
    NAS-Port = 10122
    Acct-Status-Type = Start
    Acct-Delay-Time = 0
    Acct-Session-Id = “224602551”
    Acct-Authentic = RADIUS
    Calling-Station-Id = “6307362368”
    Called-Station-Id = “7653150”
    Framed-Protocol = PPP
    Framed-IP-Address = 192.168.135.129
```

```
Sat Feb 15 07:50:55 1997
    User-Name = “t031982”
    NAS-IP-Address = 144.158.43.168
    NAS-Port = 10122
    Acct-Status-Type = Stop
    Acct-Delay-Time = 0
    Acct-Session-Id = “224602551”
    Acct-Authentic = RADIUS
    Acct-Session-Time = 10
```

Using the runtime SQL Database

7

This chapter introduces the runtime database and its tables and provides information to create user profiles in the database using a utility to interact with the database. Backing up the SQL database, both on-line and off-line, is also described.

.

Tables in the NavisRadius database

Typically, most users of NavisRadius use the native NavisRadius database and the user profiles stored in the `users` file. Some users however, need to use an SQL database as the user profile database. NavisRadius users have the option of storing user profiles and accounting outputs in a default runtime SQL database that is included with NavisRadius

NavisRadius ships with a runtime SQL database embedded with the Radius server through ODBC. This is the SQL Anywhere database from Sybase and includes tables created for storing authentication profiles of users and logging accounting information in a dedicated format, as provided by NavisRadius.

The NavisRadius database includes two tables: authentication and accounting. The tables and information in them, is available in the file create_tables.sql in the UNIX /`etc/raddb` directory or the Windows NT `c:\Ascend\NavisRadius\raddb` directory. The tables are created from `create_tables.sql`. The steps to create the database are described below:

1 CD to < ins.dir\raddb\sqlany>

2 dbinit NavisRadius.db
Start “dbeng50 –n ANRdb NavisRadius.db”
“isql read ../create_tables.sql”

3 Exit.

4 Move NavisRadius.* to <ins.dir\raddb>

Note: Please refer to Appendix D, “SQL script for authentication and accounting table,” and Appendix E, “radodbc.map file,” on the Authentication and Accounting Tables SQL Scripts.

To have the database engine start automatically when the NavisRadius servers operating system boots, do the following:

HP-UX - edit `/sbin/init.d/S810radius`
and uncomment “DBENGINE=yes” and comment out “DBENGINE=no”

Solaris - Edit `/etc/rc2.d/S81radius`
and uncomment DBENGINE=yes and comment out DBENGINE=no

NT - The engine starts automatically when the database is accessed.

Introduction to RunSQL

The RunSQL program is a utility provided with NavisRadius to be used for minor database table selections and modifications. It also supports the basic NavisRadius database table selections and modifications required to create user profiles in the database. Its input is a file containing RunSQL commands. The file can be created in a standard editor, such as notepad in Windows or a UNIX file.

To use “runsql,” you need to do the following:

On HP/UX, set the SHLIB_ PATH to include

```
<ins.dir\raddb\sqlany>
```

On Solaris, set the LD_LIBRARY_PATH to include

```
<ins.dir\raddb\sqlany>
```

Commands are either keywords, or actions. Below is a description of the keywords and actions used with this utility:

RunSQL keywords

Table 7-1. RunSQL keywords and their descriptions

Keyword	Description
ACTION=	Specifies which action to perform for the following lines in the file. The Action type must appear after the keyword. For example, ACTION=INSERT
TABLE=	Specifies which table to perform the action on.

Table 7-1. RunSQL keywords and their descriptions

Keyword	Description
KEY=	Specifies that the following column is a key and will be used in an SQL *where* clause. The KEY command is only used within rows and is available only for the UPDATE and SELECT actions.
#	A comment delimiter. Allows a user to add comments. All comments must start off the line. There can be no inline comments.

RunSQL Actions

Table 7-2. RunSQL actions and their descriptions

Action	Description
INSERT	Inserts data into a given table name.
UPDATE	Updates rows for a given table name.
DELETE	Deletes rows for a given table name.
SELECT	Selects columns from a given table.
SELECT_NONULL	Same as select, but columns with null data will not be returned.

Formatting the rows

All rows, except the SELECT action, are formatted in the following way:

```
column name>='<value>',etc ...
```

All columns must end in a comma. All rows must end with a return.

Example:

```
ACTION=INSERT, TABLE=accounting,
uname='Bill Smith',nasipaddr='199.169.30.1',
```

uname is an Accounting table column and 'Bill Smith' is the value to be inserted. All values need to be enclosed in single quotes. There is no error checking.

The SELECT action uses the following format:

```
<column name>,<column name>,etc ..
```

Again, all columns end in a comma. There is no value needed here.

Example:

```
ACTION=SELECT, TABLE=accounting,
uname,nasipaddr,
```

The output of the select action is placed in the file RunSQL.OUT.

Command Line Arguments:

To execute the RunSQL utility, an input parse file is mandatory. This file contains all the commands and actions that the user requests to perform. The default user id is 'dba', the default password is 'sql' and the database name is 'ANRdb'. Any one of these defaults can be changed by using the appropriate command line arguments.

Usage:`runsql <input file name> [-U <user id> -P <password> -<db name> ]`

runsql can use “-” or “/” to specify command line arguments

Example:

```
runsql input1.txt -U dba -P welcome -N newdb
```

Outputs

All errors are sent to the RunSQL.log file. All select actions are sent to the r RunSQL.out file.

Example Parse File

```
#
## First obtain the user name and nas ipaddress
#
ACTION=SELECT, TABLE=authentication,
uname,nasipaddr,
#
## Now truncate the accounting table and insert two rows
#
ACTION=TRUNCATE TABLE=authentication,
    uname='Jane Doe  ',nasipaddr='199.169.30.4',
    uname='Delete Me ',nasipaddr='199.169.30.5',
# Now remove the 'token' delete row
ACTION=DELETE, TABLE=authentication,
uname='Delete Me ',nasipaddr='199.169.30.5',
#
## Now insert more rows
#
ACTION=INSERT, TABLE=authentication,
    uname='Bill Smith',nasipaddr='199.169.30.1',
    uname='Herb Johnson    ',nasipaddr='199.169.30.2',
    uname='Ian Marcus   ',nasipaddr='199.169.30.3',
    uname='    ',nasipaddr='199.169.30.5',
#
## Update the accounting table where nasipaddres = some
value
#
```

```
ACTION=UPDATE, TABLE=authentication,

uname='Mr. Bill Smith  ',nasipaddr='255.255.255.0',KEY=nasi-
paddr='199.169.30.1',

uname='Dr. Herb Johnson  ',nasi-
paddr='255.255.255.255',KEY=nasipaddr='199.169.30.2',

#

## Now reselect the user name and nas ipaddress

#

ACTION=SELECT, TABLE=authentication,

uname,nasipaddr,

# Done!
```

Using ISQL to access NavisRadius

ISQL is an SQL Anywhere utility that helps to check the contents of the authentication and accounting tables that are contained in the database.

To invoke ISQL input at the command line,

C:\install dir\raddb\sqlany

C:\Ascend\NavisRadius\raddb\sqlany\isql

This starts the ISQL utility. Next, select the connect option from the commands tab at the top. This prompts the ISQL login window. The ISQL login screen asks for the following information and you need to input the DBA name, password and the database file as follows:

User ID:

Password:

Connection Name:

Database file:

Server:

Start Line:

Note: The above prompts are displayed if the database engine is not running. If the database engine is running, you will be promted only for your User ID and Password.

dba is the default for the DBA name and ANRdb for the database file. Use them to access the ISQL sections for further action.

At the command section of the ISQL screen, you can input the SQL statements that you want to use. For example:

Select* from authentication

(press EXECUTE for NT, F9 FOR HP-UX or Solaris)

The information, in this case the authentication table and its entries, is displayed on the data screen.

To exit from ISQL while using NT, select, *Exit* from the file. For HP-UX or Solaris, press F10, then select *File* ---> *Exit.*

Backing up the NavisRadius database

This section describes how to perform an off-line and on-line backup. It also describes restoring NavisRadius (SQL Anywhere database.)

Off-line Backup

To perform an off-line backup, the NavisRadius database (SQL Anywhere) should not be running. The database can be copied to any standard backup media or directory designated for backups.

Shutdown the database by entering the following command:

HP-UX - `HP/sbin/init.d/S810radius stop`
Solaris - `sol/etc/rc2.d/S81radius stop`
NT - `Stop` NavisRadius (database will shutdown automatically in about two minutes.)

Copy the database file NavisRadius.db to the specified backup media or directory.

The NavisRadius.db file is located under the database directory that you specified when you installed the NavisRadius server. For example, if you set the database directory as etc/raddb/ANRdb, you would copy etc/raddb/ANRdb/NavisRadius.db to a backup location.

A transaction log file is also written by SQL Anywhere and can be used for recovery purposes. This may be also backed up. Do *not delete* this file.

On-line Backup

Executing the dbbackup command can perform the on-line backup while the database is running. This provides a snapshot of a consistent database while it is being modified by other users.

Example:

Backup the NavisRadius.db on an NT server, stored in c:\sqlany\NavisRadius.db to a directory e:\backup using user ID dba and password SQL using the following command.

```
dbbackup -c “uid=dba; pwd=sql;

dbf=c:\sqlany\NavisRadius.db” -d e:\backup
```

Option `-d` indicates that the database is backed up, not the transaction log. If you specify -t, the transaction logs are backed up. If neither `-d` nor `-t` is specified, both are backed up.

Restoring the database

The files need to be restored from the backup created by the dbbackup program. You do the restore by copying one or both of the files according to the conditions that follow:

1. If both the files are required to be restored, replace them with the backup copies.
2. If the database file is corrupted, replace the corrupted file with the backed up file and include changes in the transaction log into the database file.

```
 cd <name of the database directory>
dbeng50 NavisRadius.db -a NavisRadius.log
```

3 If the transaction log is corrupted, replace it with the backed up log and do the following:

```
 cd<name of the database directory>
dbeng50 NavisRadius.db -f
```

NavisRadius configuration changes

Implementing the IP address allocation with radipad, and tracking state of user connections with radstated, has to be configured as indicated below while using the NavisRadius database for check and reply items.

The following examples elaborate on the configuration of this functionality with user profiles stored in the NavisRadius authentication table. For a detailed description of these features please refer to Chapter 8, “Configuring advanced features.”

Example 1

Configure all users who belong to realm ascend.com to use the ODBC interface for authentication of the NavisRadius SQL database, session limits and group limits.

In the users file configure:

```
DEFAULT Authentication-Type=Realm


radstate-hosts Password=”NAS”, Service-Type=Outbound
        Ascend-Assign-IP-Server = 168.30.15.7
```

In the authfile configure

```
ascend.com FILE ascend
```

In the ascend.users file configure:

```
DEFAULT Authentication-Type=ODBC,
   Server-Name=”aac:Authentication:dba/sql”, Session-
   Limit=2,
   Group-Session-Limit=30, Group-Limit-Name=”ascend”
```

```
    <all check items>
```

If no Group-Limit-Name is given, the realm name is taken as the default (in this case ‘ascend.com’)

Example 2

Configure all users who belong to a DNIS 1111 to use the ODBC interface to authenticate the NavisRadius SQL database, session limits and group limits.

In the users file configure:

```
DEFAULT Authentication-Type=DNIS-Realm


radstate-hosts Password=”Ascend”, Service-Type=Outbound

    Ascend-Assign-IP-Server = 168.30.15.7
```

In the authfile configure:

```
1111 FILE 1111
```

In the 1111.users file configure: DEFAULT Authentication-Type=ODBC,

```
Server-Name=”aac:Authentication:dba/sql”, Session-Limit=2,
Group-Session-Limit=30, Group-Limit-Name=”ascend”
<all check items>
```

Group-Limit-Name has to be included in the profile when using DNIS-Realm.

Example 3

Configure a carrier to use group limits and an ISP to use session limits. This example assumes that the realm ‘ascend’ is proxied to ISP.

Carrier Proxy Server

In the users file configure:

```
radstate-hosts Password=”Ascend”, Service-Type=Outbound
```

```
    Ascend-Assign-IP-Server =168.30.15.7
```

```
DEFAULT Authentication-Type=Realm
```

In the authfile configure:

```
ascend.com FILE ascend
```

In the ascend.users file configure:

```
DEFAULT Authentication-Type=RADIUS,
Server-Name=”ISP_RADIUS-Server-Name”, Group-Session-Limit
=30, Group-Limit-Name=”ascend”
<All check items>
```

ISP Authentication Server

Similar to example 1 or 2 (it depends on whether the user needs to refer to Realm/ DNIS-Realm)

Example 4

Configure dynamic IP address allocation for all users who belong to a realm ‘ascend.com’ and who use ODBC authentication to NavisRadius SQL database.

In the users file configure:

```
DEFAULT Authentication-Type=Realm

global-pool-ppp1 Password = “ascend”, Service-Type = Out-
bound
    Ascend-IP-Pool-Definition = “10.1.0.73 3”

radipa-hosts Password=”Ascend”, Service-Type=Outbound
    Ascend-Assign-IP-Server = 168.30.15.7
```

In the authfile configure:

```
ascend.com FILE ascend
```

In the ascend.users file configure:

```
DEFAULT Authentication-Type=ODBC, Server-Name=”aac:Authenti-
cation:dba/sql”
    Ascend-Assign-IP-Pool = 65535,
    Ascend-Assign-IP-Global-Pool = “global-pool-ppp1”
```

Example 5

Configure dynamic IP address allocation for all users who belong to a DNIS 1111 who use ODBC authentication to NavisRadius SQL database.

In the users file configure:

```
DEFAULT Authentication-Type=DNIS-Realm


radipa-hosts Password=”Ascend”, Service-Type=Outbound
    Ascend-Assign-IP-Server= 168.30.15.7


global-pool-ppp1 Password = “ascend”, Service-Type = Out-
bound
    Ascend-IP-Pool-Definition = “10.1.0.73 3”
```

In the authfile configure:

```
1111 FILE 1111
```

In the 1111.users file configure:

```
DEFAULT Authentication-Type=ODBC, Server-Name=”aac:Authenti-
cation:dba/sql”
    Ascend-Assign-IP-Pool = 65535,
    Ascend-Assign-IP-Global-Pool = “global-pool-ppp1”
```

Configuring advanced features

8

This chapter explains Ascend NavisRadius support for advanced features and the configuration guidelines.

Authentication features

Proxy RADIUS based on the dialed telephone number

You can use the telephone number a user dials to set up proxy RADIUS authentication if you make two changes in the users file and authfile entries that identify a user as a member of a realm. This is possible because a new value has been created for the Authentication-Type attribute. The new value DNIS-Realm. DNIS, or Dialed Number Information Service, enables the telephone network to pass the telephone number that the user dials to the call's destination.

This feature is based on NavisRadius's ability to group users with similar characteristics in a bloc called a realm. Typically, you create realms for users who are members of the same work group or are authenticated by the same kind of token card server. You can identify members of typical realms by an email-like extension added to their user names, such as jsmith@central. Users whose profiles contain the Authentication-Type = DNIS-Realm check-item, do not need a special name. NavisRadius can identify the user group they belong to by the telephone number they use to dial for access.

This feature is designed to proxy user authentication requests from one authentication server to another in a configuration where a carrier leases ports to multiple ISPs. The carrier assigns each ISP a hunt group on the MAX, MAX TNT, which is an agent of the carrier's NavisRadius server. The server provides the MAX, MAX TNT's authentication requests to the ISP's authentication servers, based on the hunt group telephone number dialed by each ISP's users.

For example, consider the actions that occur when the carrier's MAX, MAX TNT receives calls on the hunt groups assigned to the ISPs that lease the carrier's service. Further, each ISP is assigned 200 slots associated with the hunt groups 555-2000, 555-3000, and 555-4000. The ISPs' users call the hunt group number assigned to their ISP.

When the carrier's MAX, MAX TNT receives a call from an ISP's end-user, it sends an Access-Request to the carrier's NavisRadius server. The request contains the number the user dialed. The number is the value of the Access-Request's Called-Station-Id attribute (attribute number 30). The carrier's NavisRadius server searches its authfile for a realm whose name is the same as the telephone number in the Called-Station-Id attribute. The carrier's server then determines which ISP's authentication server can authenticate the user and sends it the Access-Request. When the ISP's server determines the user 's authentication status, it sends its response to the carrier's NavisRadius server. The MAX, MAX TNT receives the authentication results from the carrier's NavisRadius server, acting as the ISP's server's proxy.

Configuring NavisRadius for proxy RADIUS by DNIS-Realm

The following examples illustrate the users file and authfile entries that enable NavisRadius to proxy RADIUS authentication requests based on a user's name and a number a user dials for access.

If you want to set up proxy RADIUS authentication based on a telephone number, the DEFAULT user profile is the best user profile for adding the Authentication-Type =DNIS-Realm check-item because your objective is to match a number instead of a particular name. You might also create authfile entries for DEFAULT and NULL realms, respectively, to specify where to send authentication requests when the Called-Station-Id value is an unknown number, or when an Access-Request does not contain the Called-Station-Id attribute.

Standard proxy RADIUS authentication

The first step to enable proxy RADIUS authentication is to create a user profile in NavisRadius's users file. The profile can be that of a particular user or the DEFAULT user, and the profile must contain the check-item Authentication-Type = Realm. The second step is to add the name of the user's realm to the list of realms in NavisRadius's authfile file. The line containing the realm name must also include the name of the host that receives the Access-Request packet from the NavisRadius server acting as a proxy. Following are lines from a profile in the users file and a realm entry in the authfile file that together enable RADIUS proxy authentication:

Users file entry:

DEFAULT Authentication-Type=Realm

This is the first line of the DEFAULT profile. Any attributes and values that appear on the first line of a profile are authentication check-items. The first entry on the line indicates this is the DEFAULT user and the second indicates that any name that matches DEFAULT, rather than a particular user name, will be authenticated as a member of a realm.

Authfile entry:

ascend.com RADIUS radius.ascend.com

This line in the authfile file creates a realm named ascend.com. Another RADIUS server on the host named radius.ascend.com, authenticates the realm's members. The three components of the line are the entries for the Realm, Type, and Realm/DNS address fields.

Proxy RADIUS by DNIS-Realm

You perform steps that are very similar to those described in the previous section when you enable proxy RADIUS authentication, based on the user's access telephone number. However, in the users file profile, make the value of the Authentication-Type check-item DNIS-Realm instead of Realm, and replace the realm name in the authfile entry with the telephone number a user dials for access.

This causes the RADIUS server to treat the Called-Station-Id telephone number in the router's Access-Request like a realm name. The server searches the authfile for a realm name that matches the telephone number and the matching entry provides the name or address of the user's authenticating server. You can begin the authfile realm entry with an asterix (*). NavisRadius interprets this as a wildcard symbol so a telephone number will match even if it is preceded by an area code or other digits.

Following are examples of users file and authfile file entries that enable proxy authentication based on the value of the Called-Station-Id attribute value.

Users file:

```
DEFAULTAuthentication-Type=DNIS-Realm
```

Authfile entry:

```
*5551212 RADIUS radius.ascend.com
```

Displaying alternate sets of user messages during ACE authentication

A message catalog is a set of messages which can include user prompts. This new feature enables NavisRadius to display messages from an alternate message catalog for a user authenticating via an ACE server. Previously, during ACE authentication, NavisRadius could only display English language messages it stored as static character arrays.

Background

When users authenticate with ACE, NavisRadius must display messages which prompt the user to enter a response. International companies require NavisRadius servers which can display these messages in many languages. NavisRadius can now display messages in a language other than English if you create a new message catalog containing the messages and define an environment variable which describes the new catalog's location. You can also create message catalogs in different languages and use the procedure described in "Setting environment variables for aac_ace" to find and load the catalog which provides the correct messages for the users you want NavisRadius and ACE to authenticate.

NavisRadius is programmed to use the message catalog in a file named aac_ace, so it searches for that file when you start the NavisRadius server. The user who starts the NavisRadius server can define the location of aac_ace in an environment variable called AAC_MESSAGES, or in one of several environment variables related to the Catoctin operating system's catopen( ) function.

Caution: NavisRadius and the catopen( ) function only use the environment variables of the user who starts NavisRadius.

The NavisRadius log captures information about the catalog which has been opened and retains an entry such as:

```
securid_init() Using message catalog aac_ace in locale en_US
for user prompts
```

Creating a message catalog

You create message catalogs with your system's gencat command line utility. The gencat utility transforms the message catalog's source file into a message catalog binary file. The source file is a text file. Use NavisRadius's example message catalog source file in the directory /etc/raddb/aac_ace.ex as a guide for creating your own source file. The order of the file names after the gencat command may be source file followed by catalog file or vice versa depending on the operating system. The source file may have an extension such as .msg and the catalog file may have an extension such as .cat.

Following is an example of a gencat command to transform a file called newmsgs.msg into a message catalog file called german.cat:

```
gencat newmsgs.msg german.cat
```

Where NavisRadius searches for the aac_ace file

NavisRadius first searches for **aac_ace** in the location defined by the AAC_MESSAGES environment variable. If you want to set the aac_ace file's location in AAC_MESSAGES, you must create that environment variable because it is not created by the operating system. If you do not create and define AAC_MESSAGES, NavisRadius calls on the system's catopen( ) function to determine where the aac_ace file is located. Most systems provide an environment variable called NLSPATH and catopen( ) might first search for NLSPATH. The NLSPATH environment variable defines the Native Language Source path. If NLSPATH is not defined, catopen( ) looks in a default location which varies from system to system.

Setting environment variables for aac_ace

The environment variable ACC_MESSAGES provides the simplest way to define the location of aac_ace. If you want to keep all matters relating to NavisRadius in one location, the user who starts NavisRadius can create AAC_MESSAGES and make the environment setting the NavisRadius data directory. The data directory contains files such as users, clients, authfile and dictionary. The default data directory is /etc/raddb. Following is an entry for setting AAC_MESSAGES to the default data directory:

```
AAC_MESSAGES=/etc/raddb
```

If you want the catopen( ) function to find aac_ace via the NLSPATH environment variable, place the message catalog in a directory such as /nlslib. The catopen( ) function looks for a file name with the extension .cat, which is passed to the system in the name argument. Create an entry similar to the following where %N.cat is aac_ace.cat because %N equals the name of the message catalog passed by NavisRadius in the name argument:

```
NLSPATH=/nlslib/%N.cat
```

You can include in the NLSPATH other environment variables such as locale and LC_MESSAGES. Following is an example of such an entry for NLSPATH and a explanation of its effect:

```
NLSPATH=/nlslib/%L/%N.cat:/nslib/%N/%L
```

This entry causes catopen( ) to first look for the message catalog aac_ace.cat in the path named `/nslib/{LC_MESSAGES}/aac_ace.cat.` {LC_MESSAGES} tells catopen**( )** to use the value of another environment variable called LC_MESSAGES. If it doesn't find the message catalog there, catopen( ) looks for it in `/nslib/aac_ace.cat/{LC_MESSAGES}.`

If you want the catopen() function to use a system default location you must determine what the default environment variable is for the server's operating system. It is, for example, the default for the Solaris system is `/usr/lib/locale.`

ACC_MESSAGES

Specifies the location of the NavisRadius message catalog named `aac_ace`.

Support for backup ACE slave server

NavisRadius sends Access-Requests to a backup ACE server when it cannot contact the primary ACE server to authenticate SecureID users. The primary ACE server is the master server and the secondary ACE server is the slave server.

Background

When a user’s profile contains the check-item attribute/value pair Authentication-Type = ACE, NavisRadius contacts an ACE server and acts as that server’s agent, passing the ACE server’s challenge to the user and the user’s response to the ACE server. The user generates a response by entering a PIN into a SecureID token and obtaining a code that matches a code the ACE server expects to receive.

A file on the NavisRadius server called sdconf.rec includes the ACE server’s location and identifies NavisRadius as an ACE agent. You create this file when you install the ACE server software and must copy the file to the NavisRadius server. The sdconf.rec file also contains information about another ACE server NavisRadius can contact if the primary, or master, ACE server is unavailable. The secondary ACE server is referred to as the slave ACE server. The slave ACE server receives its user information from the master ACE server.

Configuring sdconf.rec

The “ACE/Server Installation and Configuration Guide” explains how you configure the master and slave ACE server entries in the sdconf.rec file. Once configured, copy the file to the NavisRadius server, in the `<install\dir\raddb>` directory.

Using Server-Name to specify authentication server and Kerberos realm

You can enter the Server-Name attribute in a user profile to specify the name of a RADIUS server or Kerberos realm if you cannot enter a value for the server or realm in the authfile Realm/DNS/File field. This enables you to specify an authentication server or Kerberos realm for a realm’s users.

Background

User profiles in a RADIUS users file, including a prefix.users file, can contain an Authentication-Type attribute/value pair that indicates the user’s authentication is performed by another authentication server, such as another RADIUS server, a

TACACS or a Kerberos server. The authfile file also contains entries such as DEFAULT_RADIUS_SERVER and DEFAULT_TACACS_SERVER that specify the locations of these servers.

Prior to the introduction of the Server-Name attribute there was no way to specify an alternative to the default authentication servers listed in the authfile. The Server-Name attribute enables you to enter an attribute in a user profile that specifies the location of an alternative to servers such as the DEFAULT_RADIUS_SERVER.

Using the Server-Name attribute in a realm’s user profiles

You can enter the Server-Name attribute in a realm’s user profiles to bypass the limitation of the Realm/DNS/File field entry described above. The Server-Name attribute’s value is a string. The attribute is a check-item you must enter on the first line of a user profile, along with the Authentication-Type attribute that corresponds with the value of the Server-Name attribute. For example, if you want to identify a RADIUS server to match a user profile that contains the attribute/value pair Authentication-Type = RADIUS, then create a user profile like the following:

Example:

```
realmuser Authentication-Type = RADIUS,

Sever-Name=’’radius.realm.com’’, Group-Limit-
Name="group2",

   Group-Session-Limit=25
   Service-Type = Framed,
   Framed-Protocol = PPP,
   Ascend-Route-IP = Route-IP-Yes,
   Ascend-Metric = 2,
   Framed-Routing = None,
   Ascend-Assign-IP-Pool = 2
```

Regular expressions in authfile realm names

This new feature enables you to insert a a set of alphanumeric variables within an authfile entry. One of the variables may match a character in a specific position in a realm name when NavisRadius searches the authfile during authentication. An entry with the regular expression can be a match for multiple realm names. Regular expressions are more specific than “wild cards” in realm names.

Overview of regular expressions

The server’s operating system supports regex syntax. With the addition of the regular expression feature, you can use this syntax when you create an entry in an authfile and want to specify different characters which can appear in a position in a realm name. NavisRadius searches an authfile when the value of the Authentication-Type attribute in a user profile is Realm or DNIS-Realm. The value that NavisRadius searches for appears in the Access-Request the NAS sends the NavisRadius server. The value is that of the User-Name attribute or the Called-Station-ID attribute.

Though regular expression syntax is powerful, you should use it judiciously. Each regular expression must be compiled and NavisRadius cannot search the authfile for entries which match the regular expression syntax as fast as it can search for matching wildcard entries.

When you should use regular expression

You should use regex syntax when you create realm names based on the user profile check-item, Authentication-Type=DNIS-Realm. For example, regular expression variables can represent a group of numbers which match the fifth number in the telephone number. The following table illustrates two realm names which include regular expressions. The first entry makes good use of the specificity of regular expressions in a DNIS-Realm name. The second entry, which affects a text name, is less successful.

.

Table 8-1. Authfile entries using the regular expressions feature

authfile entry	**Matches these realm names**	**Does not match these realm names**
+5551[234]12	5551212 5551312 5551412	5551512 5551612 5551712
+n[oae]rthern.sales	northern.sales. narthern.sales. nerthern.sales.	southern.sales. western.sales. easter.sales.

Note: The wildcard feature provides a better solution if you want one authfile entry to match names such as northern.sales.com, western.sales.com and eastern.sales.com. A wildcard might also be a better solution if you want to match a wide variety of numeric possibilities in a telephone number. Compare the entries in Table 8-1 with the entries in Table 8-2.

Table 8-2. authfile entries using the wildcard feature

authfile entry	**Matches these realm names**	**Does not match these realm names**
*.sales.com	northern.sales.com western.sales.com eastern.sales.com	northern.office.com western.sales sales.com
*5551212	6145551212 2165551212 7035551212	6145551213 2165551222 7035551012

Creating authfile entries with regular expressions

You can create authfile entries with the text editor for the authfile. You must enter values for the Realm Name and Type fields. Other entries, such as Protocol, are optional. The Realm/DNS/File field is optional, but can be required, based on your entry in the Type field. For example, if you enter RADIUS in the Type field, NavisRadius will be a proxy server for a remote RADIUS server and you must enter the remote server's name or IP address in the Realm/DNS/File field.

Text editor

1 Open an authfile located in the NavisRadius data directory.

Note that NavisRadius may have more than one authfile.

2 Enter a line containing values for the Realm Name and Type fields.

Place the entry near the end of the file, unless most searches are for entries which contain regular expressions.

Regular Expression

Description: This feature determines whether the value of a User-Name attribute or a Called-Station-ID attribute passed to NavisRadius in a NAS Access-Request matches an entry in an authfile.

Usage: Regular expressions can be part of an authfile entry representing the name of a realm or the telephone number of DNIS-Realm. The regular expression enables one entry to be a match for multiple names or telephone numbers.

Example: Following are authfile entries which include a regular expression:

```
#RealmAliasesProtocol TypeREALM/DNS/PREFIX

+jsmith[123]RADIUSrad.ascend.com

+5551[234]12RADIUSrad.ascend.com
```

Dependencies: Following are dependencies for inserting regular expressions in authfile entries:

- The entries must begin with a plus sign (+).
- The variables in the regular expression's set of characters must be enclosed in square brackets ([]).

- The set of variables which appear between the square brackets must not be separated by spaces, tabs, or other white space.
- Regular expression realm names are case sensitive.

Note: The order of authfile entries is important when you use DNIS-Realms. The authfile will be searched each time the NAS receives a call. If entries which include regular expressions are not the most common entries, you can reduce the work for radiusd by placing them at the end of the file.

See Also: authfile, Called-Station-ID, DNIS-Realm, realm, User-Name.

Restrict authentication by service-type or framed protocol with Prohibit

The Prohibit attribute limits the kinds of connections a user can make. Eleven new types of connections have been added to the list of the Prohibit attribute's values. The names of the original five values have been changed. The format of the Prohibit attribute/value pair is now Prohibit=Prohibit-value.

Overview

Prohibit is an authentication check-item. Administrators can add Prohibit to a user profile so the user cannot be authenticated for a specific type of connection. The list of connections from which the user can be prohibited has been increased from five to sixteen. The list now corresponds with the values of the Service-Type and Framed-Protocol attributes and the value all, which prevents the user from being authenticated for any connection. The original list of Prohibit values included:

Dumb, now called **Prohibit-Login**

- **PPP**, now called **Prohibit-PPP**
- **SLIP**, now called **Prohibit-SLIP**
- **Auth-Only**, now called **Prohibit-Authenticate-Only**
- **All**, now called **Prohibit-All**

New Prohibit values include:

- **Prohibit-None**
- **Prohibit-Callback-Login**
- **Prohibit-Callback-Framed**
- **Prohibit-Outbound-User**
- **Prohibit-Administrative-User**
- **Prohibit-Shell-User**
- **Prohibit-Authenticate-Only**
- **Prohibit-Callback-Admin-User**
- **Prohibit-ARA**
- **Prohibit-Gandalf**
- **Prohibit-Xylogics**
- **Prohibit-Framed**

Prohibit (1028)

Description: Prohibit is an authentication check-item. The Prohibit check-item specifies a type of connection which NavisRadius will not authenticate. Several of the values which you can assign the Prohibit attribute perform administrative functions, such as Administrative-User and Authenticate-Only.

Usage: All check-items must appear in the first line of a user profile. You can enter more than one Prohibit attribute in a user profile, provided each is in the first line. You can use the values Prohibit-Authenticate-Only and Prohibit-Administrative-User to perform administrative functions. You can use the value Prohibit-Authenticate-Only to confirm that a user profile will successfully authenticate a user, since no authorization reply-items are returned in the Access-Accept packet. You can also use the value Prohibit-Administrative-User to confirm that a server is on-line. Prohibit-Administrative-User translates to a server-status code and does not prohibit a connection.

RADIUS example

This profile prevents any authenticated connection for a user who is in arrears on service payments.

```
deadbeat Password="randomstring", Prohibit=Prohibit-all,
   Service-Type=PPP,
   Framed-IP-Address=137.157.8.8,
...
```

This user profile prevents the authentication of outbound connections for localuser:

```
localuser Password="randomstring", Prohibit-Outbound-User,
   Service-Type=Outbound-User,
   Framed-IP-Address=137.157.8.8,
   ...
```

Global IP pools, radipad and radstated

Proxy RADIUS in a Global IP Address Pool environment/ Radipad

Users who authenticate via proxy RADIUS can be assigned IP addresses from Global IP Address Pools. In proxy RADIUS the NAS IP address allocation and release messages that are passed from the proxy server to the authenticating server must contain information about the user's name, or the user's realm or the user's DNIS-Realm so the proxy server knows where to forward the messages. Features added to MAX and MAX TNT enables them to place User-Name and Called-Station-ID attributes in the messages. This NavisRadius feature enables the proxy RADIUS server to use these attributes to determine which authenticating server should receive the messages.

Overview

A Global IP Address Pool is a centralized pool of IP Addresses shared by more than one Network Access Server. A NAS temporarily needs an IP address from a pool to configure a WAN interface when an authenticated user's profile contains the reply-item Ascend-Assign-IP-Pool=65535.

radipad is the daemon that allocates IP addresses from Global IP Address Pools. A NAS configuring an interface requests a pool IP address from radipad, then informs the daemon when the IP address is no longer needed. The NAS communicates with radipad through NavisRadius, which passes NAS messages to the daemon.

When RADIUS finds Ascend-Assign-IP-Pool=65535 in an authenticated user’s profile, it alerts the NAS by sending that reply-item in its Access-Accept message. The NAS responds with an Ascend-IP-Allocate, message (type 50.) The message prompts NavisRadius to look in the user’s profile for the Ascend-Assign-IP-Global-Pool attribute. NavisRadius adds the value of that attribute to the NAS’s Ascend-IP-Allocate message and passes the message to radipad. The value of the attribute specifies the pool from which radipad is to choose an IP address the NAS can use to configure the connection.

When the connection closes, the NAS sends an Ascend-IP-Release type message to radipad via RADIUS and the daemon releases the IP address for assignment when it receives the message.

To permit proxy RADIUS, MAX, MAX TNT sends the Called-Station-ID attribute in its Ascend-IP-Allocate message and the User-Name or Called-Station-ID attribute in its Ascend-IP-Release message.

Radipad communicates with NavisRadius on tcp port 9992. To change the default, edit your systems “services” file and specify “radipad <port#>/tcp.” NavisRadius and radipad must both be restarted in order for these changes to take effect.

Radipad Configuration

Radipad is available on both the UNIX and NT versions of NavisRadius.

Configuring the users file

This following example describes a generic configuration defining multiple pools, followed by a sample user configuration.

```
#USERS FILE
radipa-hosts Password = "ascend" , Service-Type = Outbound
   Ascend-Assign-IP-Server = <ip address>,

   global-pool-max Password = "ascend", Service-Type = Out
bound
```

```
   Ascend-IP-Pool-Definition = "172.31.179.1 6"
   Ascend-IP-Pool-Definition = "172.31.179.65 62"
   Ascend-IP-Pool-Definition = "172.31.179.129 50"
   Ascend-IP-Pool-Definition = "172.31.179.193 8"
globalpooluser Password = "test"
   Service-Type = Framed,
   Framed-Protocol = PPP,
   Ascend-Assign-IP-Global-Pool = "global-pool-max"
   Ascend-Assign-IP-Pool = 65535,
```

Startup radiusd and radipad processes

Edit the 'services' file in your operating system and specify:

```
radipa-startup 9992/tcp # start radipad
```

Configuration changes on the MAX and the TNT

Max

Do not define default IP pools in Ethernet>Mod Config>Wan Options

```
Set Ethernet>Mod Config>Auth. > Auth Pools = Yes
```

TNT

Disable the IP Pools under the IP-Global profile (in pool-base-address and assign-count)

```
Set External-Auth>Rad-Auth-Client>Auth-Pool = Yes
```

Tracking the state of user connections with radstated

NavisRadius includes `radstated`, a daemon that tracks the state of user connections, enabling you to limit the simultaneous connections users can make. The RADIUS daemon contacts radstated before accepting a user whose profile includes the Simultaneous-Use attribute. This attribute's value equals the user's maximum number of simultaneous connections.

Overview

NavisRadius authenticates users and authorizes their connections, but radiusd, the RADIUS daemon, does not keep track of user connections. The radstated daemon does track the state of user connections, so you can use it in conjunction with radiusd to restrict the number of connections a user makes. You can install radstated on the NavisRadius server or on another host that communicates with the NavisRadius server. radstated contacts radiusd when users request authentication for more connections than they are authorized to make.

`radstated` receives requests to check users' session limits on port 9993 and responds via the same port number. The daemon receives accounting information about the status of users' connections at port 9994.

To change the default, edit your operating systems "services" file and specify: "radstated <port #>/tcp" and "radstate-acct <port #>/udp."

NavisRadius and radstated must both be restarted in order for this changes to take effect.

Note: For both radipad and radstated, a single instance of radipad /radstated can service multiple NavisRadius Clients. It may be located on the same machine (or not) as a NavisRadius server.

Creating profiles supporting radstated connection tracking

You must create a special user profile for radstate-hosts in the users file before you start radstated. The radstate-hosts profile must contain at least one Ascend-Assign-IP-Address reply-item attribute. The attribute's value is the IP address of a system running the radstated daemon and the entry tells NavisRadius where radstated is running. You can include multiple Ascend-Assign-IP-Address attribute/value entries in the radstate-hosts profile if the radstated daemon is running on more than one host. Following is an examples of a radstate-hosts user profile which includes the locations of two radstated daemons:

```
radstate-hosts Password = “Ascend”, Service-Type = Outbound
   Ascend-Assign-IP-Server = 137.175.80.159
   Ascend-Assign-IP-Server = 137.175.80.220
```

The profile of each user whose connections are limited must contain the Simultaneous-Use attribute. The attribute must be entered on the profile's first line because Simultaneous-Use is an authentication check-item. The Simultaneous-Use attribute's value is an integer which equals the number of connections NavisRadius can authenticate for the user.

Note: Do not include the Simultaneous-Use attribute in profiles of users whose connections are not limited.

Following is the profile of a user who can only make five simultaneous connections.

```
Rossellini Password = “cittaaperta”, Simultaneous-Use = 5
    Service-Type = Framed,
    ...
```

Configuring radiusd when radstated is tracking the state of connections

When radstated is tracking the state of user connections, you can start radiusd with the -Sa or -Sf command line option. These options permit radiusd to correctly handle authentication requests if it cannot send packets to, or receive packets from, the host running radstated. By default, radiusd fails authentication requests for users with simultaneous-use limitations when radstated cannot be contacted because you have not entered -Sa or -Sf as one of radiusd's command line options.

Sa

Description: The `Sa` option causes radiusd to allow users to pass the limit check when the daemon cannot communicate with radstated.

Usage: Enter on the command line when starting the RADIUS daemon.

Example: radiusd -Sa

Dependencies: Use only if the radstated daemon is tracking the state of user connections.

Sf

Description: The `Sf` option causes radiusd to fail the limit check when it cannot communicate with radstated.

Usage: Enter on the command line when starting the RADIUS daemon.

Example: radiusd -Sf

Dependencies: Use only if the radstated daemon is tracking the state of user connections.

Virtual Private Hardware (or group session limits)

NavisRadius's Virtual Private Hardware (VPH) feature enables you to create user groups and to assign each of the groups a specific number of ports to which they can connect. Assigning a group a specific number of ports restricts the number of users in a group that can simultaneously connect. VPH works with radstated, RADIUS State Daemon that tracks the state of user connections.

Background

Virtual Private Hardware is essentially a group session limit, used to limit the number of dial-up sessions a group of users may use. You can use the Virtual Private Hardware feature when you want to limit a number of users by means other than the traditional hunt group and physical port limits. For example, carriers who lease blocks of remote Access server ports to Internet Service Providers, can use VPH to restrict the number of ports an ISP's customers use to the number of ports that the ISP has leased.

How the Virtual Private Hardware feature works

radstated, the external RADIUS State Daemon, already permits NavisRadius to limit the number of times a user can connect across a full network. VPH utilizes radstated in a similar, but not identical way. After users are assigned to groups, radstated can keep track of connected users, the locations to which the users dial in, and the group to which the users belong.

One RADSTATE daemon should track all the group's users to effectively limit the number of the group's connections. If you run a separate daemon in each Point of Presence (POP) you can only limit the group's connections by POP, rather than by the total connections of the group across the network.

VPH requires that the groups' users have a group session limit entry in their user profiles. Placing a group session limit in a user's profile does not prevent you from entering a session limit in the profile, so radstated can track the user's connections to see if he surpasses individual or group limits.

Following is the sequence of events that occurs when radiusd and radstated are running and a user requests a connection from a NAS. The NAS sends an Access-Request to radiusd and radiusd looks up the user's entry in the users file. radiusd returns an Access-Reject to the NAS if the user's authentication information does not match the entry. If the user authentication information matches the entry and the entry contains a user session limit or group session limit, then a request is sent to radstated to ascertain if the user's current number of connections has reached or exceeded the value of either attribute in his user profile.

If the session limit has been reached then the Access-Request is rejected. If the session limit has not been reached then the connection is accepted and the reply-items in the user's entry are returned to the NAS.

Session limits are part of the authorization process that follows authentication. In a proxy RADIUS situation, this means that the home RADIUS server will be contacted first to authenticate the user before the session limit is checked to see if the user has exceeded their limit. If the session limit has been reached then his Access-Request will be rejected, even though the home RADIUS server has authenticated the user.

Once the user is connected, the NAS sends an Accounting Start message to radiusd. If a radstate-hosts is configured for radiusd, that daemon forwards the information on user, group, and location (NAS-IP-Address, NAS-Port, NAS-Port-Type) to radstated whether or not a session limit has been specified for the user. If the user has a session limit, this information will be stored and the session counts for the user's name and group will be incremented. If the user does not have a session limit, radstated checks its state table and removes and discards any stale information.

When the user disconnects, the NAS sends an Accounting Stop message to radiusd. If the user has a session limit, the information is forwarded to radstated. The session counts for the user’s name and group are decremented and the state record will be removed.

If a NAS reboots and sends either an Accounting-Off or Accounting-On message to radiusd, the daemon passes this information to radstated and the session information for all users who were connected to the NAS is removed and the session counts updated.

Configuring the Virtual Private Hardware feature

The VPH feature requires changes to both radiusd and radstated. You must update both programs in order to make use of the feature. The new radstated can understand both old and new radiusd messages.

Setting per-group or per-user session limits for a user requires that the NAS be configured to send both authentication and accounting messages to radiusd.

The VPH feature is a RADIUS server feature so you do not need to change the NAS software. A user or group limit also does not have any effect on the packets passing between the servers. The limits are simply values that must be checked before authorization can complete. The group name and limit do not get put into the packets as a reply item, nor are they forwarded between RADIUS servers.

Configuring radiusd to identify radstated

To configure NavisRadius’s VPH feature, you must configure radiusd so that it knows the location of the RADSTATE daemon and you must create user profiles for the members of the groups that you want radstated to track.

To configure radiusd to know radstated’s location, you must place a user entry called radstate-hosts in the RADIUS daemon’s users file. The radstate-hosts entry must contain the Ascend-Assign-IP-Server attribute and the attribute’s value must be the IP address of the host running radstated.

Example:

```
radstate-hosts Password = "Ascend", Prohibit = Prohibit-All
    Ascend-Assign-IP-Server = 10.0.0.2
```

Configuring user profiles of group users

You assign users to a group by entering check-items in their NavisRadius user profiles that identify them as members of the group. Two new attributes have been created to be used as the check-items for setting group limits in a user profile. The new attributes are Group-Limit-Name (1040) and Group-Session-Limit (1041). Both the Group-Limit-Name and the Group-Session-Limit attributes are user profile check-item attributes because they must be checked to authenticate a user's request for access. They must appear in the first line of a user's profile. Examples of the use of Group-Limit-Name and Group-Session-Limit attributes in user profiles are in the attribute descriptions in the sections "Group-Limit-Name," and "Group-Session-Limit."

Using new attributes that support VPH

The value of the Group-Limit-Name is a string that represents the group to which a user belongs. You must enter the attribute in any group member's user profile and the value for each member of the same group must be the same. For example, the profiles of all users who are members of a group named North must have the attribute/value pair Group-Limit-Name=”North”.

Note: Group-Limit-Name is not the same as the attribute called Group-Name.

The value of the Group-Session-Limit attribute is an integer that represents the number of connections the members of the group can have to the network tracked by the RADSTATE daemon. If the total number of sessions for the group exceeds the value of Group-Session-Limit, a user who requests access to the network is not permitted to connect.

You do not have to enter a Group-Session-Limit in the profile of each member of the group. A group user whose profile does not contain the Group-Session-Limit attribute will not be checked against the group limit when attempting to connect to the network. In effect this permits this user to connect to the network regardless of the fact that the group limit has been exceeded. A connection that is permitted a user whose profile does not contain the Group-Session-Limit attribute does count against the group's session limit and therefore does affect the ability of other group users whose profiles do contain the Group-Session-Limit attribute to connect.

Configuring VPH for use with REALMs or DNIS-REALMS

You can use the VPH feature with a REALM or a DNIS-REALM if the RADIUS daemon can find the realm's DEFAULT user profile and that profile contains the Group-Limit-Name and/or Group-Session-Limit attributes. To permit this to happen you must create a prefix.users file for the realm where you can place the DEFAULT user profile and you must change the realm's authfile entry so radiusd can find the file.

A *prefix.users* file is a special users file that is only associated with a particular realm. The variable prefix in prefix.users is the name of the realm whose users' profiles are in the file. NavisRadius checks the prefix.users file instead of the users file when the authentication type field is FILE in the realm's authfile entry and that entry contains the name of the prefix.users file.

For example, the following authfile entry for the little1 realm indicates that the authentication type is RADIUS and that the RADIUS server which authenticates the realm's users is radius.little1.com.

Example:

```
little1              RADIUS      radius.little1.com
bigco1 (bigco1.com)  RADIUS      radius.bigco1.com
admin                ACE
local                UNIX-PW
```

To enable the use of group session limits for users in the little1 realm, you can create a users file called *little1.users*. Then create a *DEFAULT* user profile such e as the following in the *little1.users* file.

Example:

```
DEFAULT Authentication-Type = RADIUS, Server-Name = 
radius.little1.com,

Group-Limit-Name = "little1", Group-Session-Limit = 25
```

Then edit the realm's authfile entry as shown below:

Example:

```
little1                 FILE     little1
bigco1 (bigco1.com)     RADIUS   radius.bigco1.com
admin                   ACE
local                   UNIX-PW
```

Sharing group session limits across realms

You can cause multiple realms to share the same group session limits if you enter the same Group-Limit-Name and Group-Session-Limit attribute/value pairs in each realm's DEFAULT user profile.

Combining DNIS authentication and group session limits

You can use group session limits in combination with two stage DNIS authentication on Ascend equipment to reject users before a call is answered by the MAX, MAX TNT. You can also do this in a proxy RADIUS situation so that the home RADIUS server is not contacted for the initial call.

For example, create the following user profiles in the little1.users file described in "Configuring VPH for use with REALMs or DNIS-REALMS."

Example:

```
5551212 Password="Ascend-DNIS", Group-Limit-Name="little1",
Group-Session-Limit= 25
   Ascend-Require-Auth=Require-Auth
DEFAULT Authentication-Type = RADIUS, Server-Name =
=radius.little1.com, Group-Limit-Name = "little1", Group-
Session-Limit = 25
   Service-Type = Framed,
   Framed-Protocol = PPP,
   Ascend-Route-IP = Route-IP-Yes,
   Ascend-Metric = 2,
   Framed-Routing = None,
   Ascend-Assign-IP-Pool = 4
```

The above example will check the limit both when the call is initially received and when the user authenticates. Because of the delay between the two authentication steps, it is possible in this situation that a user's call will be accepted but the user will be rejected because they went over the limit (the count of users on-line is not updated until the Accounting Start message is received). To prevent this, you can remove the Group-Session-Limit from the DEFAULT entry with the risk that the user may exceed the group session limit in some cases where they would not have done so previously.

Group-Limit-Name

Description: Specifies the name of the group to which a user belongs. The Group-Limit-Name may be applied to the DEFAULT user.

Usage: Enter the attribute/value pair in the first line of the user's profile in the users file.

Example*:*

```
user1 Password="test", Group-Limit-Name="group1", Group-
Session-Limit=100
   Service-Type = Framed,
   Framed-Protocol = PPP,
   Framed-IP-Address = 192.0.2.1,
   Framed-IP-Netmask = 255.255.255.0

DEFAULT Authentication-Type=UNIX, Group-Limit-Name="group2",
Group-Session-Limit= 25
   Service-Type = Framed,
   Framed-Protocol = PPP,
   Ascend-Route-IP = Route-IP-Yes,
   Ascend-Metric = 2,
   Framed-Routing = None,
   Ascend-Assign-IP-Pool = 2
```

Dependencies: The Group-Limit-Name must be included as a check-item in the user profile of any user that belongs to a group.

See Also: Group-Session-Limit, Group-Name

Group-Session-Limit

Description*:* Specifies the total number of active sessions a group can have across the network served by a RADSTATE daemon. Not a required check-item in the user profiles of all members of the group. More connections by a group's users than are specified by the attribute's value are permitted if the user profiles of individual members of the group contain the Group-Limit-Name attribute but do not contain the Group-Session-Limit attribute. May be entered in a user profile that contains the Simultaneous-Use attribute that limits the number of simultaneous sessions an individual can have on the network.

Usage*:* Enter the attribute/value pair in the first line of a group member's user's profile in the users file if you want the user's request for access to be constrained by the group's total of allowable connections.

Example*:*

```
user238 Password = "test", Simultaneous-Use = 2, Group-
Limit-Name = "group3", Group-Session-Limit = 200
    Service-Type = Framed,
    Framed-Protocol = PPP,
    Ascend-Route-IP = Route-IP-Yes,
    Ascend-Metric = 2,
    Framed-Routing = None,
    Ascend-Assign-IP-Pool = 3
```

Dependencies: Must be used with the Group-Limit-Name attribute.

- Can use with REALMs or DNIS-REALMS if the authfile authentication type field is changed to FILE and you create a prefix.users file that contains a DEFAULT user profile that has the Group-Limit-Name, Group-Session-Limit attributes.

See Also: Group-Limit-Name, Simultaneous-Use

radstated option to disable SNMP verification

The RADSTATE daemon can be started with an option that prevents radstated from using SNMP to verify the number of a user's simultaneous connections. Use radstated's -A option to force the daemon to rely on RADIUS Accounting-Request messages to track the state of a user's simultaneous connections.

Overview

The Simultaneous-Use attribute can be entered in a user’s profile if you need to limit the number of simultaneous connections the user can open. The attribute’s value equals the number of sessions the user is permitted. If the user reaches her session limit then radstated sends SNMP messages to the Network Access Servers where the daemon believes the user’s connections exist.

SNMP objects are in a vendor-specific MIB and radstated can be used in a mixed vendor environment where all the NAS machines might not support the MIB. If radstated can not use SNMP to verify a user’s connection count you must be able to turn off the daemon’s SNMP messages.

-A option

Description: The `-A` option disables the ability of radstated to send SNMP messages.

Usage: Enter the `-A` option on the command line when starting the RADSTATE daemon.

Example: radstated -A

Specify SNMP read community string global value with radstated

Starting radstated with the -C option enables you to specify the value of the SNMP read community string which radstated sends to NASes when the daemon attempts to verify how many connections a user has open.

Overview

The value of the Simultaneous-Use attribute is the number of sessions that a user is permitted to have open at one time. When a user whose sessions are limited by the Simultaneous-Use attribute requests a connection, and radstated thinks the user has reached the maximum number of connections he is permitted, then radstated sends SNMP messages to the Network Access Servers where the daemon believes the connections exist.

To be able to read the SNMP objects radstated must know the value of the READ community string on all the NAS. Every NAS that radstated must contact must use the same read community string.

If the value of the community string on each NAS has been changed the radstated -C option enables you to update the value of the string that the daemon will send so that it matches the string that is currently accepted by all of the NAS.

radstated -C command line option

Description: The -C option enables you to change the value of the community string in the SNMP messages that radstated sends to a NAS when it attempts to verify the number of a user's simultaneous connections.

Usage: The -C option takes one argument. The argument must be present if the -C option is used. Enter the -C option and a string value on the command line when starting the RADSTATE daemon. If no -C option flag is put on the command line the value of the community string that radstated sends defaults to the string public.

Example: `radstated -C readstring`

Tunnel attribute support

Tunneling attributes

NavisRadius supports attributes that cause compulsory tunneling of data over authenticated connections. Multiple tunnel configurations can be defined by tunnel attributes in a user profile and each tunnel attribute can be tagged to associate it with a particular tunnel configuration. NavisRadius also supports a flag that identifies a NAS as a machine that does not recognize tunnel attribute tags.

Overview

RADIUS attributes that implement mandatory tunneling for authenticated connections have been described in RADIUS draft documents. The tunnel attributes described in the draft documents enable administrators to use RADIUS user profile authorizations to force tunneling of data transmitted and received over an authenticated user's connection.

The RADIUS tunnel drafts mirror the RADIUS functions described in the RADIUS authentication and accounting documents, RFC 2138 and RFC 2139, respectively.The titles of the RADIUS tunnel draft documents are:

- "RADIUS Attributes for Tunnel Protocol Support"
- "RADIUS Accounting Modifications for Tunnel Protocol Support"

Some of the proposed tunneling attributes that an administrator can enter in a user profile enable the administrator to specify such things as the protocol that creates the tunnel, the medium over which the tunnel is created, and the tunnel endpoints. Table 7-6 contains six tunnel attributes described in the RADIUS tunnel authentication and accounting drafts that have been added to the NavisRadius dictionary because the attributes are supported by Ascend MAX and Ascend TNT products

.

Table 8-3. Tunnel attributes added to the NavisRadius dictionary

Attribute Type	Attribute Name (#)	Attribute Description
Tunnel auth/acct	Tunnel-Type (64)	Tunneling protocol to be used, ex. L2TP, ATMP.
Tunnel auth/acct	Tunnel-Medium-Type (65)	Tunnel transport medium to be used, ex. IP, HDLC.
Tunnel auth/acct	Tunnel-Client-Endpoint (66)	Address of the tunnel initiator.

Table 8-3. Tunnel attributes added to the NavisRadius dictionary

Attribute Type	Attribute Name (#)	Attribute Description
Tunnel auth/acct	Tunnel-Server-Endpoint (67)	Address of the server end of the tunnel
Tunnel acct	Tunnel-ID (68) (Acct-Tunnel-Connection in draft)	Identifier assigned to the session to uniquely identify a tunnel session for auditing purposes.
Tunnel auth	Tunnel-Password (69)	Password to authenticate to a remote server.

Using the Tag field to identify attributes

In addition to describing RADIUS tunnel attributes, the RADIUS tunnel authentication draft also introduces a new field for dictionary file entries. The new field is called the Tag field and it can be used to mark attributes that can be encrypted, or that can be linked to other attributes in a user profile. The Tag field appears at the end of entries for STRING and INTEGER type attributes in a RADIUS server’s dictionary file.

The values for the Tag field that can appear in a dictionary file entry are TAGGED and SALTED. Attributes in a dictionary that include the value TAGGED can be identified as a part of a tunnel configuration in a user profile. Entries that include the value SALTED can be encrypted using the Salted-Encryption algorithm that is described in another RADIUS draft document titled, “Salt-Encryption of RADIUS Attributes”. Both values may apply to one attribute, such as the Tunnel-Password attribute.

All the attributes that have been added to the NavisRadius dictionary file to support tunneling include the value TAGGED in the Tag field. The Tunnel-Password attribute that was added also includes the value SALTED in the Tag field.

Following is an example of a dictionary file entry that includes the TAGGED value in the Tag field.

```
ATTRIBUTE Tunnel-Type 64 integer (*, 0, TAGGED)
```

Configuring the Tag field in a user profile

When configuring the Tag field value in a user profile, enter the attribute and its value, followed by a colon (:) and the value of the Tag field.

Following is an example of a user profile that includes tunnel attributes and Tag field values that link the attributes that are part of a specific tunnel configuration.

```
Username Password = “secret value”
    Tunnel-Type = L2TP : 2,
    Tunnel-Medium-Type = IP : 2,
    Tunnel-Server-Endpoint = “lap-aydin” : 2,
    Tunnel-Type = ATMP:3,
    Tunnel-Medium-Type = IP:3,
    Tunnel-Server-Endpoint = “lap-aydin”:3
```

Identifying NAS that do not support the Tag field

Some NAS might not support the new Tag field for attribute dictionary file entries. The Tag field has been designed so that if the attribute’s Tag is not configured by the RADIUS server, the attribute’s value is backward compatible with the original dictionary entry format for RADIUS attributes. The RADIUS server can identify a NAS that does not support the Tag field if the NAS’ clients file entry includes the flag NOTAGS. For example, a RADIUS server would recognize that a NAS called MAX, one that is identified by the following clients file entry, does not support the Tag field.

```
MAX-one sha red-secret type = Ascend:NAS+NOTAGS
```

If the NOTAGS flag is added to a client’s entry in the clients file and an attribute in a user profile is marked as a TAGGED or SALTED attribute in the dictionary, then radiusd does not fill the Tag field with a meaningful value when it sends the client the user profile attributes. Instead, the server sets the value of the Tag field to zero for INTEGER attributes it sends to a NOTAGS client, and does not insert the Tag field at all when it sends STRING attributes to those clients, unless the

STRING attributes are SALTED. radiusd sets the Tag field value to zero if user profile STRING attributes are tagged as SALTED.

In addition, if a client is flagged as a NOTAGS type client, radiusd prunes multiple instances of a single TAGGED attribute that appear in a user profile so that Access-Accept messages the server sends the client contain only one copy of the attribute.

Note: clients file entries for all Ascend MAX and TNT units should contain the NOTAGS flag because the units do not support the attribute Tag field. However, MAX and TNT units can use the Salted Tunnel-Password attribute and they expect, but ignore, the value in the Tag field

Note: During configuration of the NAS, the VSA option on Ascend NAS gear has to be set for authentication and accounting.

Tunnel attributes

Following are descriptions of the six attributes that have been added to the NavisRadius *dictionary* file.

Tunnel - Type (64)

Description: Description: The Tunnel-Type attribute specifies the protocol that the tunnel initiator should use to create a tunnel. The attribute can appear in Access-Request, Access-Accept and Accounting-Request messages.

Note: A tunnel initiator is not required to implement the tunnel type that appears in an Access-Accept message. If the tunnel initiator does not know or support the tunneling protocol represented by the numeric value of the Tunnel-Type attribute that it receives, then the initiator behaves as though the message were an Access-Reject message

Usage: The value entered for this attribute is an acronym that is associated with a protocol. Table 8-4 lists the twelve Tunnel-Type attribute tunneling protocols and the acronyms that can be entered as values for the attribute.

Table 8-4. Tunneling protocols and their values in the Tunnel-Type attribute

Tunneling protocol	Tunnel-Type value
Point-to-Point Tunneling Protocol	PPTP
Layer Two Forwarding	L2F
Layer Two Tunneling Protocol	L2TP
Ascend Tunnel Management Protocol	ATMP
Virtual Tunneling Protocol	VTP
IP authentication Header in Tunnel Mode	AH
IP-in-IP Encapsulation	IP-IP
Minimal IP-in-IP Encapsulation	MIN-IP-IP
IP Encapsulating Security Payload in Tunnel Mode	ESP
Generic Routing Encapsulation	GRE
Bay Dial Virtual Services	DVS

Example: Following is an example that displays how a Tunnel-Type attribute that has been tagged to a specific compulsory tunnel configuration, might appear in a user profile.

```
Username Password = “secret value”
   Tunnel-Type = L2TP : 2,
   Tunnel-Medium-Type = IP : 2,
   Tunnel-Server-Endpoint = “lap-aydin” : 2
```

See Also: “Tunnel-Medium-Type (65).”
“Tunnel-Client-Endpoint (66).”
“Tunnel-Server-Endpoint (67).”

"Tunnel-ID (68)."
"Tunnel-Password (69)."

Tunnel-Medium-Type (65)

Description: The Tunnel-Medium-Type attribute specifies the transport medium that can be used to create a compulsory tunnel described by the tunneling attributes in a user profile if the tunneling protocol can operate over more than one transport medium, such as L2TP. The Tunnel-Medium-Type attribute can appear in Access-Request and Access-Accept messages.

Usage: Enter Tunnel-Medium-Type as an authorization attribute in a user profile. The attribute's value is one of the one of those listed in the Assigned Numbers standard, RFC 1700 under the heading "Address Family Numbers". A relevant sample of the list includes:

- IP (IP version 4)
- IP6 (IP version 6)
- NSAP
- HDLC
- BBN 1822
- 802 (includes all 802 media plus Ethernet "canonical format")
- E.163 (POTS)
- E.164 (SMDS, Frame Relay, ATM)
- F.69 (Telex)
- X.121 (X.25, Frame Relay)
- IPX
- Appletalk
- Decnet IV
- Banyan Vines
- E.164 with NSAP format subaddresses

Example: Following is an example that displays how a Tunnel-Medium-Type attribute that has been tagged to a specific compulsory tunnel configuration might appear in a user profile.

```
Username Password = “secret value”
   Tunnel-Type = L2TP : 2,
   Tunnel-Medium-Type = IP : 2,
   Tunnel-Server-Endpoint = “lap-aydin” : 2
```

See Also: “Tunnel -Type (64).”
“Tunnel-Client-Endpoint (66).”
“Tunnel-Server-Endpoint (67).”
“Tunnel-ID (68).”
“Tunnel-Password (69).”

Tunnel-Client-Endpoint (66)

Description: The value of the Tunnel-Client-Endpoint attribute is the address of the initiator end of the tunnel. The attribute may appear in Access-Request and Access-Accept messages to indicate the address from which a new tunnel is to be initiated. The attribute should also be included in Accounting-Request packets if those packets contain Acct-Status-Type attributes which have the value of Start or Stop.

Usage: Enter the Tunnel-Client Endpoint attribute in user profiles. The value of the attribute is a string that is dependent on the value of the Tunnel-Medium-Type attribute that is also entered in the user profile. If the value of the Tunnel-Medium-Type is IP or IP6, the string is either a fully qualified domain name or a dotted-decimal IP address. If the value of the Tunnel-Medium-Type is not IP or IP6, the value of the Tunnel-Client-Endpoint attribute in the user profile is a tag that refers to configuration data local to the RADIUS client. The configuration data must describe the interface and medium-specific address to use as the Tunnel-Client-Endpoint value.

Example:

```
Username Password = “secret value”
   Tunnel-Type = L2TP : 2,
   Tunnel-Medium-Type = IP : 2,
   Tunnel-Client-Endpoint = “MAXaccess” : 2
```

See Also: “Tunnel -Type (64).”
“Tunnel-Medium-Type (65).”
“Tunnel-Server-Endpoint (67).”

"Tunnel-ID (68)."
"Tunnel-Password (69)."

Tunnel-Server-Endpoint (67)

Description: The value of the Tunnel-Server-Endpoint attribute is the address of the server end of the tunnel. The attribute may be included in an Access-Request packet. It must be included in Access-Accept packets the RADIUS server sends the NAS. The attribute should also be included in Accounting-Request packets if the packets contain an Acct-Status-Type attribute which has the value of Start or Stop and the packets pertain to a tunneled session.

Usage: Enter the Tunnel-Server Endpoint attribute in user profiles. The value of the attribute is a string that is dependent on the value of the Tunnel-Medium-Type attribute that is also entered in the user profile. If the value of the Tunnel-Medium-Type is IP or IP6, then the string is either a fully qualified domain name or a dotted-decimal IP address. If the value of the Tunnel-Medium-Type is not IP or IP6, then the value of the Tunnel-Client-Endpoint attribute in the user profile is a tag that refers to configuration data local to the RADIUS client.

Example:

```
Username Password = “secret value”
    Tunnel-Type = L2TP : 2,
    Tunnel-Medium-Type = IP : 2,
    Tunnel-Server-Endpoint = “lap-aydin” : 2
```

See Also: "Tunnel -Type (64)."
"Tunnel-Medium-Type (65)."
"Tunnel-Client-Endpoint (66)."
"Tunnel-ID (68)."
"Tunnel-Password (69)."

Tunnel-ID (68)

Description: The Tunnel-ID attribute indicates the identifier assigned to the session. The attribute appears in Accounting-Request packets which contain

Acct-Status-Type attributes that have values of either Start or Stop. Together, the Tunnel-ID, Tunnel-Client-Endpoint and Tunnel- Server-Endpoint attributes provide a means to uniquely identify a tunnel session for auditing purposes.

Note: The Tunnel-ID attribute is the Acct-Tunnel-Connection attribute that is described in the “RADIUS Accounting Modifications for Tunnel Protocol Support” draft.

See Also: “Tunnel -Type (64).”
“Tunnel-Medium-Type (65).”
“Tunnel-Client-Endpoint (66).”
“Tunnel-Server-Endpoint (67).”
“Tunnel-Password (69).”
Tunnel-Password (69)

Description: The Tunnel-Password attribute contains a password that enables a user to authenticate a compulsory tunnel connection with a remote server. The attribute’s password value is hidden during transmission because it is encrypted by means of the Salted Encryption mechanism. Tunnel-Password attributes in user profiles can be tagged so that they are associated with specific compulsory tunnel configurations in the profile. The Tunnel-Password attribute can only appear in an Access-Accept message.

Note: Salted Encryption is described in a draft called “draft-ietf-radius-saltencrypt-00.txt”. The Salted Encryption mechanism permits any RADIUS attribute to be encrypted. Attributes that can encrypted by the mechanism are tagged in the RADIUS dictionary by the addition of the SALTED flag at the end of the attribute’s dictionary entry.

The Salted Encryption mechanism adds a unique two-octet Salt value to each attribute to be encrypted. The Salt is concatenated with the secret shared by the RADIUS server and the remote server and the Request Authenticator from the corresponding Access-Request. The result is input to the MD5 digest which produces an initial 16-byte XOR value that is unique for each encrypted attribute in a RADIUS transaction. The initial and subsequent XOR values are used to encrypt the attribute’s payload.

Usage: Enter the Tunnel-Password attribute and its value in a user profile. The value of the attribute is a string surrounded by quotes.

Example:

```
Username Password = “secret value”

Tunnel-Type = L2TP : 2,
Tunnel-Medium-Type = IP : 2,
Tunnel-Client-Endpoint = “MAXaccess” : 2,
Tunnel-Password = “2435saltedpass7980” : 2
```

See Also: “Tunnel -Type (64).”
“Tunnel-Medium-Type (65).”
“Tunnel-Client-Endpoint (66).”
“Tunnel-Server-Endpoint (67).”
“Tunnel-ID (68).”

Other features

Using time of day restrictions in user profiles

The Time-Of-Day attribute has been added to the NavisRadius dictionary to enable you to restrict access based days of the week and the times of the day.

Overview

A user requesting authentication must provide information that matches the value of each check-item in his user profile. When the profile contains the Time-Of-Day check-item RADIUS will authenticate the user and authorize the user’s connection if the day of the week and the time of the day match the value of the Time-Of-Day check-item.

The values you can enter for the Time-Of-Day attribute are based on the days of the week and a 24 hour clock cycle. You can also enter three inclusive values, Wk, Any, and Never.

Following are examples of ways you can use the Time-of -Day attribute to control user access.

- Restrict access to the period between 5 PM and 8 AM.

- Prohibit weekend access.
- Prohibit all access to the network.

Users receive a Reply-Message if they attempt to connect during a day or time when the value of the Time-Of-Day check-item indicates that they should not be able to connect. The message says:

```
Invalid connection time.
```

Configuring user profiles with Time-Of-Day authentication

The Time-Of-Day attribute is a check-item with a value that is a string. Following is the format of the string. The string must be enclosed in quotes.

```
“keyword hhmm-hhmm”
```

You must enter a value for the string’s keyword, using one of the values in Table 8-5, or a combination of the values in the table.

Table 8-5. Keyword values and what they represent

Value	**Representing**
Mo	Monday
Tu	Tuesday
We	Wednesday
Th	Thursday
Fr	Friday
Sa	Saturday
Su	Sunday
Wk	Any weekday

Table 8-5. Keyword values and what they represent

Value	Representing
Any	Any day of the week
Never	No day or time

The Time-Of-Day value’s hhmm-hhmm string is optional. You can use it to specify a time range that modifies the value’s keyword. The optional string refers to a pair of 24 hour clock hour and minute settings, such as 0830-1730, which represents the period from 8:30 AM to 5:30 PM.

The range specified by the pair of hour and minute settings always refers to the hours in a single day. The time frame cannot encompass two consecutive days, such as Monday night and Tuesday morning. Therefore the optional value 1730-0830 means that the user will not be provided access between the hours of 8:30 AM and 5:30 PM on the day specified by the keyword. The value 1730-0830 represents two time periods on the same weekday during which access is permitted. One time period is between the hours 5:30 PM and 12:00 midnight. The other time period is between the hours of 12:00 midnight and 8:30 AM on the same day.

You can enter multiple keywords by using a vertical bar (|). The vertical bar means or and you can use it to separate one keyword and its associated pair of hour and minute settings from the next. If any one of the keywords and its qualifiers matches, then authentication is permitted to proceed.

Table 8-6 contains examples of Time-Of-Day attribute/value pairs and the access they permit.

Table 8-6. Time-of-day attributes values and the access they permit

Attribute/value pair	Access permitted
Time-of-Day=Any 1700-0800	Connect any day between 5 PM and 8 AM.
Time-of-Day = MoTuWeThFr	Connect any weekday, any time.

Table 8-6. Time-of-day attributes values and the access they permit

Attribute/value pair	Access permitted
Time-Of-Day = Wk1700-0800|Sa|Su	Connect any weekday between 5PM and 8 AM and all day on Saturday and Sunday.

Time-Of-Day

Description: The Time-Of-Day attribute enables you to control when a user can connect. The attribute is a user profile check-item. The days and times that a user requests authentication must match those specified in the Time-Of-Day check-item's value or the RADIUS server will not authenticate the user or authorize a connection.

Usage: On the first line of a user profile enter the attribute with a value for the keyword and optionally, a value for the hhmm-hhmm modifier of the keyword. The value for a keyword can be any of those shown below, or a combination of the values. You can optionally use a vertical bar (|) to separate multiple values for the attribute's keyword. The attribute's value must be enclosed in quotes.

- Mo
- Tu
- We
- Th
- Fr
- Sa
- Su
- Wk (Same as MoTuWeThFr)
- Any (Same as MoTuWeThFrSaSu)
- Never

Following is user profile for a user who is only allowed to connect during off-peak hours, during week days and any time on the weekend. The user is assigned a dynamic IP address each time he calls.

Example:

```
cheapuser Password = “hardtoknow”, Time-Of-Day = “Wk1800-
0800|SaSu”
   Service-Type = Framed,
   Framed-Protocol = PPP,
   Ascend-Assign-IP-Pool = 1,
   Ascend-Route-IP = Route-IP-Yes,
   Framed-Routing = None
```

Following is an example of a user profile for a user who has been denied access at any time.

```
blocked Password = “donttrust”, Time-Of-Day = “Never”
```

User profile entries that authorize IPX packet filters

Ascend-Data-Filter and Ascend-Call-Filters attributes containing configurations for packet filters that control IPX traffic can be entered in NavisRadius user profiles. When the user is authenticated the IPX filters in the profile are sent to MAX and TNT routers.

Overview

Ascend MAX and TNT units support IP, GENERIC, and IPX packet filters. The filters define specific types of packets and also specify actions the units must take when packets that match the filters’ definitions appear at one of the router’s interfaces. When the filters are active they cause the unit to examine the packet stream and the unit forwards or blocks packets as specified by the filters.

IP filters affect the way a unit handles packets that are constructed according to the Internet Protocol specifications. GENERIC filters include parameters that might match any type of packet. The MAX and TNT security and RADIUS supplements contain more information about IP and GENERIC filters.

IPX filters affect the way a unit handles Novell NetWare packets. MAX and TNT support for IPX filters was added in Release 6.0.0 and information about IPX filters is described in the units’ Release Notes.

Format of IPX entries in Ascend-Call-Filter and Ascend-Data-Filter attributes

Following are the formats for entering values for Ascend-Call-Filter and Ascend-Data-Filter attributes.

Usage: Filter entries apply on a first-match basis. Therefore, the order in which filter entries appear as values for the Ascend-Call-Filter and Ascend-Data-Filter attributes is significant. Fields shown below that are enclosed by parentheses, such as srcipxnet or dstipxsoc, are optional.

Ascend-Call-Filter

```
Ascend-Call-Filter="ipx <dir> <action>
    [srcipxnet <0xnnnnnnnn> srcipxnod <0xmmmmmmmmmmmm>
        [ srcipxsoc <cmp> <value> ] ]
    [dstipxnet <destipxnetaddr> dstipxnod <destipxnode>
        [ dstipxsoc <cmp> <value> ] ]
```

Ascend-Data-Filter

```
Ascend-Data-Filter="ipx <dir> <action>
    [srcipxnet <0xnnnnnnnn> srcipxnod <0xmmmmmmmmmmmm>
        [ srcipxsoc <cmp> <value> ] ]
    [dstipxnet <destipxnetaddr> dstipxnod <destipxnode>
        [ dstipxsoc <cmp> <value> ] ]
```

Table 8-7. IPX Call and Data filter syntax elements

Keyword or argument	Description
ipx	Keyword to designate an IPX filter.
<dir>	Direction of packets affected by filter, IN or OUT.
<action>	Action taken unit on packets matching the filter, FORWARD or DROP.

Table 8-7. IPX Call and Data filter syntax elements

Keyword or argument	Description
srcipxnet	Keyword for source IPX address. nnnn = IPX Network Number.
srcipxnod	Keyword for source IPX Node address. mmmm = IPX Node Address. **Note**: Could be FFFFF’s.
srcipxsoc	Keyword for source IPX socket address.
<cmp>	Argument indicating how to compare specified value to actual src value. One of following four values: <, >, =, !=
<value>	Socket value, in hex, against which the value in a packet is to be compared.
dstipxnet	Keyword for destination IPX address. nnnn = IPX Network Number.
dstipxnod	Keyword for source IPX Node address. mmmm = IPX Node Address. Note: Could be FFFF.
dstipxsoc	Keyword for destination IPX socket address.

Map multiple vendors' attributes and disable the mapping function

In the clients file entry of a NAS you can list multiple vendors for which NavisRadius can map vendor-specific attributes. You can also use the NOMAP flag to disable NavisRadius's vendor-specific attribute mapping function.

Overview

The type field in a NAS clients file entry is an optional field. You can enter information in the type field that defines a NAS so that radiusd can select the attributes that the NAS can understand from those that the server finds in a user profile. You can also enter flags in the type field that define other characteristics of the NAS.

Vendor-specific RADIUS attributes

The RADIUS RFC lists the attributes that can be included in RADIUS packets and associates those attributes with specific numbers. For example, the attribute Service-Type is listed as number 6 in the RADIUS RFC. RADIUS is extensible and NAS vendors can create attributes that are specifically related to their own products. The RADIUS RFC specifies that the vendors' attributes should be encapsulated within the RADIUS Vendor-Specific attribute, which is number 26 in the RFC list. However, some vendors assign numbers to the attributes they create and do not encapsulate them inside attribute number 26.

Mapping vendor-specific attributes

NavisRadius prevents attribute conflicts when it is receiving packets from a NAS by automatically mapping attributes for an Ascend Network Access Server from the RFC attribute space to Ascend's attribute space. NavisRadius can also map packets from the Ascend vendor-specific attribute space to the RFC space when it is sending packets to an Ascend NAS.

NavisRadius checks the clients file entry of NAS or PROXY machines that send or receive RADIUS packets. If the entry's type field contains a list of vendors, then NavisRadius checks the vendors file to see if attribute maps exist for those vendors. If attribute maps exist, then NavisRadius maps the attributes. If the vendors file does not contain an attribute map for a particular vendor NavisRadius cannot perform mapping.

Disabling attribute mapping

You can enter a flag in a clients file entry to disable the NavisRadius mapping function. NavisRadius will not perform attribute mapping for a vendor if the NOMAP flag is in the clients file entry, even if an attribute map exists in the vendors file.

Format of a clients file type field entry

Following is the format of a clients file type field entry that displays how lists of vendors and flags must appear. The type value that follows the colon that separates the vendors and flags lists can specify whether the client is a NAS or a PROXY, such as another RADIUS server. The value of the type option of the type field must always be entered in the flags list, although it can appear anywhere in the flags list.

```
type = vendor+vendor+...:type+flag+flag...
```

Listing multiple vendors in the clients file type field

If the RADIUS client is a NAS that can accept or send more than one vendors' vendor-specific attributes, then the NAS clients file entry must include the optional type field. The first list in the field's value must be the vendor list. Each vendor name in the list is separated from the preceding name by a plus sign (+).

For example, following is the clients file entry for a NAS that can accept attributes created by Ascend and by Microsoft.

```
Name      secret type

MAX        maxsecret type = Ascend+Microsoft:NAS
```

Listing flags in the clients file type field

A list of flags cannot precede a list of vendors in the type field and, if both lists are present, the lists must be separated by a colon (:). Individual flags must be separated by a plus sign (+). The values currently supported for flags in the clients file type field NOMAP and NOTAGS. Please refer to section, “Identifying NAS that do not support the Tag field” on page 8-32, for more information on NOTAGS.

The list of flags can also contain a value that indicates the type of RADIUS machine that sends requests to the server. The type field values for specifying the type of machine include NAS, PROXY, DAS, and FRGW.

Following is the entry for a NAS that can accept attributes created by Ascend and by Microsoft and for which NavisRadius does not map vendor-specific attributes to standard attributes:

```
Name  secret    type
MAX   maxsecret type = Ascend+Microsoft:NAS+NOMAP
```

Caution: The NOMAP flag in the clients file, must be configured to match what is set on the Ascend NAS, such as support for the VSA option.

iPass global proxying

NavisRadius supports iPass global proxying. The feature permits NavisRadius servers at remote iPass access points to forward the requests to central iPass servers. NavisRadius servers at the users’ home locations can also respond to requests from central iPass servers.This feature is not available on NavisRadius for Windows/NT

Overview

iPass is a company whose technical and operational infrastructure permits Internet access from major business capitals around the world via a local call. iPass provides this service for the dial-up subscribers and corporate customers of the service providers, technology vendors and resellers that partner with iPass.

Remote iPass access points and local Internet service providers can use NavisRadius to authenticate iPass users. Remote iPass access points can use NavisRadius iPass AATV to proxy authentication requests to an iPass central authentication daemon. Local ISPs can use the NavisRadius server’s vnas (Virtual NAS) daemon to receive authentication requests from a central iPass authentication daemon.

For example, the following actions occur when a user dials a remote iPass access point and both the remote iPass access point and the user’s organization use NavisRadius

1 The ISP’s iPass software encrypts the user’s name and password.

2 The iPass AATV on the remote ISP’s NavisRadius server securely passes the result to a central iPass server.

3 The central iPass server forwards the encrypted authentication information to the vnas daemon on the NavisRadius server.

4 The NavisRadius server sends authorization to the central iPass server which passes the authorization to the remote ISP server.

5 The remote ISP connects the user.

Installing iPass support files

The iPass global proxying feature is not available in NavisRadius for Windows/NT, but all UNIX versions of NavisRadius support the feature. The distribution’s Readme file contains instructions for installing iPass support files.

To obtain more information about iPass contact iPass Alliance Inc. at http://www.ipass.com.

Enabling iPass proxy capability

You enable the NavisRadius iPass proxy capability by entering the -i command line option for the radiusd command, as shown:

```
radiusd -i
```

Creating an Authfile entry for iPass users

The value IPASS has been created for the NavisRadius Authentication-Type attribute. Therefore IPASS can also be used as the value of the Type field in a realm’s Authfile entry All iPass users are members of the iPass roaming consortium realm.

In general make IPASS the Type value in the Authfile entry for the DEFAULT realm because a NavisRadius server at a remote iPass access point will not recognize iPass users’ realm name. Following is an example DEFAULT Authfile entry that contains the IPASS value in the Type field.

```
Realm-name Type Realm/DNS/File

DEFAULT IPASS myradiushost.mybusiness.com
```

The Authfile entry’s Realm/DNS/File field is typically the hostname of the machine on which radiusd is running.

Note: Please see Appendix B, “iPass notes,” for additional iPass information, including installation, testing and configuration guidelines.

Ascend NavisRadius files and commands

A

This chapter contains information about the Ascend NavisRadius files, commands and command options..

authfile

Name

`authfile` - Ascend Control file for mapping realms and authentication types.

Synopsis

UNIX

`/etc/raddb/authfile`

`Windows NT`

`c:ascend\NavisRadius\raddb\Authfile`

Description

The `authfile` file is installed in the Ascend NavisRadius server's default directory, unless configured differently by the system administrator. `authfile` is read by `radiusd` at startup and whenever `radiusd` receives a `HUP` signal (UNIX) or the server is refreshed (Windows NT).

`authfile` contains a list of the names of all the realms to which users seeking authentication may belong. Ascend NavisRadius refers to the list of realms in `authfile` when authenticating incoming requests from anyone offering the ***user@ realm*** patterns as a match for the Ascend NavisRadius User-Name attribute.

`authfile` may contain comments, which are indicated by leading pound sign (#) character. Comments and blank lines inserted in authfile are ignored.

`authfile` contains one line of information per realm entry. Each entry may include several, white-space delimited fields.Two fields, Realm-name and Type must appear in the entry. Other fields are dependent on entries in the required fields. See the section on Type fields for more information.

The syntax for an `authfile` entry is:

```
Realm-name [(Alias [, Alias])] [-Protocol] Type [Realm/DNS/
File] [filter-id]
```

Example:

```
umich.edu AFS-KRB UMICH.EDU

UMICH

Flatland.org (FILE) flatland
```

Realm-name field

A Realm-name entry may be any appropriate symbol or name for the realm to which the user ‘belongs.’ Note that this does not have to be a Domain Name System (DNS) host name, although it is highly recommended that the realm name match a domain name so that the user recognizes the ***user@realm*** syntax which resembles their email address.

There is also a "wild card" syntax, *.realm, which may be included as the primary (non-alias) realm name on an entry in the authfile. Its purpose is to provide a shorthand way of indicating several related realms which are to be handled by one entry (one authentication type) in the authfile.

For example, a company may have several branches, eastern.foobah.com, western.foobah.com and central.foobah.com, so that an entry *.foobah.com would "match" all three of these realms.

It is highly recommended that any such wild card entry be listed toward the end of the authfile. This allows for preceding, specific entries to override the wild card entry.

DEFAULT

A single Realm-name entry of DEFAULT may be placed at the end of the `authfile`. The field entries for the DEFAULT realm indicate how Ascend NavisRadius should handle authentication requests that include Realm-name entries that are not found in `authfile`. DEFAULT usually includes Type and Realm/DNS/File entries for a remote server. Access-requests for DEFAULT entries are sent to the remote server, which authenticates the user and sends reply-item attributes back to the local server. A user is not allowed to use the name DEFAULT as their actual realm name.

The NULL Realm-name entry is used to indicate to Ascend NavisRadius how it should handle user names that appear as *user* instead of *user@realm*.

Wild Card

The wild card syntax is `*.realm.` Wild card syntax allows you to indicate several related realms which are to be handled by one authentication Type entry in the `authfile`. For example, a company may have several branches, including `east.foo`, `west.foo` and `south.foo`. The entry `*.foo` matches all three realms. It is highly recommended that wild card entries be listed toward the middle of the `authfile` so entries for specific realms, like `east.foo` will be matched before the wild card entry like `*.foo`.

Alias (option)

The,<alias> field is an optional, comma-separated list of realm names enclosed within parentheses, which can be attached to the user name in lieu of the actual realm name. Each realm alias is equivalent to the 'main' realm name and may be provided for user convenience or other purposes, such as to save typing. For example, if the Realm- name is `foobar` and its aliases are `foo` and `bar` the user could enter `jsmith@foobar`, `jsmith@foo`, or `jsmith@bar` to match the Realm-name. Aliases are allowed for wild card entries and are interpreted as `*.alias` rather than `alias.realm` or just `alias`.

Protocol (option)

-Protocol is an optional indicator that identifies the authentication protocol which applies to the realm. The three allowable entries in the -Protocol option are `-PW`, `-CHAP` and `-DFLT`. It may be used to force the processing order of authfile entries which would otherwise be identical.

These entries are searched in order, so you may distinguish between or among otherwise identical realm entries. `-PW` and `-CHAP` stand for *Password* and *Challenge Handshake Authentication Protocol*. By default an entry applies to both Password and CHAP, but using either of the `-PW` or `-CHAP` options limits the entry to that specific protocol.

Type field

Valid authentication types include `PASSWD, UNIX-PW, RADIUS, MIT-KRB, AFS-KRB, FILE, TACACS,` and `TACPLUS,WIN/NT , and KCHAP.` These authentication types are case insensitive.

PASSWD and UNIX-PW

The PASSWORD type is the same as the UNIX-PW type. Either of these entries refers to authenticating a user by comparing his entry to his password in the UNIX password file (usually found in the /etc/password file).

RADIUS

The entry RADIUS indicates authentication is done by a remote NavisRadius server. The remote server sends reply-items, or authorization attribute/value pairs, back to its client, the local Ascend NavisRadius server. The local Ascend NavisRadius server then sends the reply-items to the NAS that originally sent the Ascend NavisRadius Access-Request packet. An Ascend NavisRadius Type entry requires a Realm/DNS/File field entry. See Realm/DNS/File below.

MIT-KRB, AFS-KRB

Either of these Type entries indicates that kerberos authentication is done at the default kerberos realm. Note that the file named `krb.conf` on your system must have valid entries for the realm. Kerberos entries require an entry in the Realm/DNS/File field. See Realm/DNS/File below.

File

The File entry indicates that Ascend NavisRadius uses a flat file lookup, searching for encrypted passwords in the `users` file. A Type field entry of File requires a Realm/DNS/File field entry. See Realm/DNS/File below.

TACACS, TACPLUS

A TACACS or TACPLUS keyword entry in the Type field indicates that authentication is done by encrypted request to a TACACS or TACACS+ server. A TACACS or TACPLUS Type entry requires a Realm/DNS/File field entry. See Realm/DNS/File below.

WIN/NT

This entry in the authentication type entry indicates that NavisRadius supports authentication of users from the Windows/NT Domain database.

ARA-DES

An ARA-DES entry requires that the Framed-Protocol entry is Ascend-ARA and this encryption must be performed by an Ascend NavisRadius server in direct communication with the NAS. It may not be performed via proxy authentication.

ACE

An ACE keyword in the Type field means that authentication involves an ACE server and a hand held SecureID device.

DEFENDER

This is similar to the ACE keyword. DEFENDER indicates that the user is authenticated by an AssureNet Defender server and SecureNet Key.

SKEY

This entry indicates that the user authenticates with a one time password method developed by Bellcore.

realm/DNS/file field

Entries in the Realm/DNS/File field are determined by the keyword entered in the Type field.

Table A-1. `Type` *field and realm/DNS/file entry relationships*

type Keywords	realm/DNS/file entries
ACE	DNS for the appropriate ACE server
AFS-KRB, MIT-KRB	Kerberos realm name
DEFENDER	DNS for the appropriate DEFENDER server

Table A-1. `Type` *field and realm/DNS/file entry relationships*

type Keywords	realm/DNS/file entries
FILE	Prefix for the realm’s `users` file in the syntax `prefix.users`. For example, the entry *`flatland`* indicates the user profiles for the realm are in the file `flatland.users` **Note:** Do not include a period at the end of the entry.
RADIUS	DNS for remote Ascend NavisRadius server
TACACS, TACPLUS	DNS for the appropriate TACACS or TACACS+ server

See also

dictionary, users, radiusd, hostname, signal, radcheck, radpwtst

clients

Name

`clients` - Ascend NavisRadius file for mapping clients to shared-secrets

Synopsis

UNIX

`/etc/raddb/clients`

Windows NT

c:\Ascend\NavisRadius\raddb\clients

Description

The `clients` file is installed in the Ascend NavisRadius server's default directory, unless configured differently by the system administrator. `clients` is read by `radiusd` at startup and whenever `radiusd` receives a `HUP` signal (UNIX) or the server is refreshed (Windows NT).

`clients` contains a list of Ascend NavisRadius's clients, such as Network Access Servers (NAS) or other RADIUS servers.

`clients` may contain comments, which are indicated by leading pound sign(#) character. Comments and blank lines inserted in `clients` are ignored.

`clients` contains one line of information per Ascend NavisRadius client entry. Each line may include up to five white-space delimited fields. The first two fields, System-name[:port] and Key must appear in the entry. The other three fields are optional.

The syntax for a `clients` entry is:

```
System-name[:port] Key Type Version Prefix
```

Example:

```
System-name[:port] Key      Type     Version Prefix

merit.edu:7       badges0  type=nas  v2      pfx

10.1.2.3:256       test    type=nas  v2      pm1
```

System-name[:port] field

This entry may be a valid Domain Name System hostname or an IP address in dotted-quad notation. The entry may also be followed by a colon (:) and a UDP/ TCP port number on the client or server. The System-name port number option overrides the default port numbers 1812 and 1813, or the ports defined by the `radiusd` `-pp` or `-qq` options. These ports define the Ascend NavisRadius and Ascend NavisRadius accounting ports, respectively. If you use the System-name port number option, the port number you enter should be for the Ascend NavisRadius port because the accounting port is assumed to be one greater than the number entered as the option.

A pair of Ascend NavisRadius clients or servers may be specified using the alternate System-name notation `name1/name2:`

```
Italia/Italy bgmsbkym type=nas
```

The alternate notation allows the same `clients` file to be distributed to physically different Ascend NavisRadius servers that have been identically configured. A request from a Ascend NavisRadius machine only matches this alternate notation if the packet's source IP address matches the IP address of `name1` or `name2` and the hostname of this server, as returned by the hostname command, matches `name1` or `name2`. You must not use the port number option if you use the alternate `name1/name2` notation.

Key field

The Key field is the encryption key or shared secret known by a Ascend NavisRadius server and a client, or by Ascend NavisRadius and another RADIUS server. The secret field may be sixteen (16) characters long.

Type field (option)

The optional Type field specifies the client's vendor name and/or the type of the Ascend NavisRadius machine sending requests to this server. The vendor is indicated by placing the vendor's name in front of the client type and separated with a colon (":") character. For example:

```
type = ascend:nas
```

If the Type field is omitted, the client type and vendor name are unspecified. Note that this may be vitally important, since the Ascend NavisRadius server only honors a vendor- specific attribute if the vendor's name is in the Type field of the associated clients file entry. Currently, Ascend NavisRadius recognizes three values for the Type field:

- NAS
- PROXY
- ASCEND

Version field (option)

The optional Version field specifies the Ascend NavisRadius version number. If this is omitted, it defaults to version one. Version one is described in the IETF RADIUS standard document. Version two is described in the IETF document titled, `draft-calhoun-enh-radius-00.txt.` Currently, `v1`, `v2` and `v3` are allowed as values in the Version field.

Prefix field (option)

The Prefix field value is a text string. It may used to select different users or authfile files for requests from the associated system. This feature allows for different RADIUS clients to share the same RADIUS server, but use different authentication databases on this server.

When you enter a string in Prefix field Ascend NavisRadius can only search for the client's realms in a specific `authfile.`The value of the Prefix field is added to the beginning of `authfile` to create the file's name. If the value of the Prefix field is `ascend,` the file that Ascend NavisRadius will search is named `ascendauthfile`. If you have entered the client's list of realms in an `authfile` named `ascend.authfile,` you must add a period at the end of the Prefix field entry so `ascend.`is the field's value. "Clients file format" on page 4-6 contains a discussion of the differences between entries in the `clients` file Prefix field and the `authfile`'s Real/DNS/File field. Both fields are used to create prefixes for file names.

See also

authfile, users, vendors, radiusd, hostname, signal, dictionary, radcheck, radpwtst

dictionary

Name

`dictionary` - translations for parsing Ascend NavisRadius requests

Synopsis

UNIX

`/etc/raddb/dictionary`

Windows NT

c:\Ascend\NavisRadiusl\raddb\dictionary

Description

The `dictionary` file is installed in the Ascend NavisRadius server's default directory, unless configured differently by the system administrator. `dictionary` is read by `radiusd` at startup and whenever `radiusd` receives a `HUP` signal (UNIX) or the server is refreshed (Windows NT).

The `dictionary` file is distributed with Ascend NavisRadius. You can edit the `dictionary` with a text editor such as vi (UNIX) or Notepad (Windows NT)

`dictionary` contains a list of Ascend NavisRadius attribute/value pair translations the Ascend NavisRadius server uses to parse incoming authentication requests and generate outgoing authorization responses.

`dictionary` may contain comments, which are indicated by leading pound sign("#") character. Comments and blank lines inserted in `dictionary` are ignored.

Each attribute's value is specified as one of four data types:

- string - 0-253 octets
- ipaddr - 4 octets in network byte order

- integer - 32 bit value in big endian order (high byte first)
- date - 32 bit value in big endian order - seconds since 00:00:00 GMT, Jan. 1, 1970

Attribute entries consist of four required fields and one optional fifth field:

```
Attribute Attribute-name Integer-encoding Type [Pruning]
```

Value entries consist of four fields

```
Value attribute-name value-name integer-encoding
```

Example:

```
Attribute Framed-Protocol 7 integer (1,0)
Value Framed-Protocol PPP 1
```

The Attribute line’s optional `Pruning` field has a unique syntax:

```
[ ( [<ack>] [ [ [, ][<nak>] ] [ [,]
[MAY\MUST\CONFIG] ] ] ) ]
```

`ack` values affect the RADIUS server’s Access-Accept replies and `nak` values affect its Access-Reject replies. If the whole expression is omitted, the default `(0,0 MAY)` is assumed. The keywords `MAY` and `MUST` are only meaningful for RADIUS versions 2 and 3. Table A-2 lists the meanings of the values for the `ack` and `nak` keywords.

Table A-2. Meanings of the ack and nak values in a RADIUS server’s replies.

Value	**Meaning**
0	No attributes of this kind are part of the final reply.
1	At most, one attribute of this kind may be part of the final reply.
*	Any number of attributes of this kind may be part of the final reply.
MUST	The NAS must reject the Access-Request if it does not understand this attribute.

Table A-2. Meanings of the ack and nak values in a RADIUS server's replies.

Value	**Meaning**
MAY	The NAS may reject, not reject, or silently discard the Access-Request.
CONFIG	This attribute is a configuration item only.The CONFIG keyword is only for the internal use of the Ascend NavisRadius server and must appear by itself (for example, `config)` at the end of the attribute line.

Vendor Specific Attributes

Ascend NavisRadius supports this syntax for handling vendor-specific attributes:

```
vendor:attribute-string
```

`vendor` is the vendor's name and `attribute-string` is a unique string for that vendor. Vendor specific attribute and value identifier strings are defined in the `vendors` file and these strings may be used in place of default attribute/value strings.

Example:

```
Ascend.attr Ascend.value 529 Ascend 26 172
```

See also

users, vendors, radiusd, signal, radcheck

dnscheck

Name

dnscheck checks forward and reverse DNS entrie**s**

Synopsis

dnscheck [-c] [-h] [-t] [-v] [-x] [host . . .]

Description

The `dnscheck` utility checks the forward and reverse DNS entries for the given host (s) or for the value returned by gethostname (2) if the host name or IP address is omitted. The host argument may be either a normal domain name such as myhost.foobah.org or an IP address in dotted-quad [123.234.123.234] form and there may be more than one such argument given.

The intent of this utility is to run it without any arguments and have it tell you if your system's concept of what it's name is matches, or is consistent with, what the DNS thinks is the name of your system. This is useful in tracking down obscure problems when configuring the clients file.

The forward and reverse information is sent to standard output followed by any information for multiple address and or multiple DNS names, one per line. If the host given is a NAME, both it and the fully qualified DNS name are printed along with its IP address. If the host given is a fully qualified DNS name which has aliases (CNAME), `dnscheck` is not able to retrieve these from the DNS.

Options

The following options are used with `dnscheck.`

Table A-3. The dnscheck options and their descriptions are listed below

Options	Description
-c	Toggles checking mode. This option defaults to ‘on.’ If this option is toggled to off, the dnscheck utility performs only one of the two checks, either forward or reverse, for the given host.
-h	Shows a usage message. <dns-name> should be checked. If no <dns-name> is given, the system hostname is used.

Table A-3. The dnscheck options and their descriptions are listed below

Options	**Description**
-t	Turns on terse mode. This option defaults to ‘off’.
-v	Shows the version information.
-x	This option is used to increase the debugging level. The debugging level is initially zero. Each -x option adds to the default, thus enabling more debugging information to be produced.

See Also: gethostname, clients

radcheck

Name

`radcheck` - determines whether an Ascend NavisRadius server is operational

Synopsis

```
radcheck [ -d dir] [ -p port] [ -r retries] [ -t timeout]
[ -x] [ -o] [ -v] servername
```

Description

`radcheck` determines if the Ascend NavisRadius server whose DNS name is entered on the command line is operational. `radcheck` may be executed on any host, even if it is not registered in the Ascend NavisRadius `clients` file. See `authfile` for more information.

If the server is operational, `radcheck` displays the following on standard output:

```
auth queue:a/b, acct queue: c/d, maxtime: t (date)
authfile: x, clients: y, users: z, date
RADIUS version ersion config codes
servername  (port-number) is responding
```

If the number of retries is greater than zero, radcheck will also display:

```
(n retries)
```

otherwise, `radcheck` displays

```
servername  (port-number) some message
```

where one of these, among others, may be some message:

```
No reply RADIUS server  <hostname> (port)
Received non-matching id in server response
Received invalid reply digest from server
```

```
   No such server:  <hostname>
```

If the sever is built with LAS (Local Authorization Server) support, the above output is preceded by the following:

```
Status: g authen, h unconfirmed, i connected, j P suspended,
k unknown
```

Also, if any token pools are configured (see tokenpool.las for more information), there will be one line for each such pool and the total number of pools displayed in the following manner:

```
some-pool-name : p/q r timestamp
```

```
number of pools: n
```

p is the total number of tokens configured in some-pool-name,

q is the current number of tokens in use

r is the token high-water-mark recorded at date and time \timestamp\fP.

Options

Table A-6 lists the options with which you can run the `radcheck` program.

Table A-4. *`radcheck` options and descriptions.*

Option	Description
`-d` dir	Allows user to specify an alternate directory instead of the default, `/etc/raddb`
-o	Specifies that the ‘old’ RADIUS packet type Access Request should be used instead of a Status-Server packet.
`-p` port	Allows user to specify an alternate port number instead of the default, 1812
`-r` retries	Allows user to specify a number of retries instead of the default, 10

Table A-4. `radcheck` *options and descriptions.*

Option	**Description**
`-t` timeout	Allows user to specify a maximum time-out different than the default, 3 seconds
`-x`	Allows user to turn on debugging output
`-v`	Prints the version of Ascend NavisRadius used to build the program

Exit Status

Table A-7 lists the `radcheck` exit status codes.

Table A-5. `radcheck` *exit status values and descriptions.*

Status	**Description**
-2	Ascend NavisRadius server response errors
-1	Local server errors
0	Normal successful completion
1	Time-out errors

See also

authfile, clients, dictionary, users, radiusd, radpwtst, radpwtst

radipad

Name

```
radipad is the RADIUS IP Address Daemon for global address
pool management
```

Synopsis

radipad [-v] [-x -p] [-d <sec>] [-n]

Description

The RADIUS IP Address Daemon (radipad) provides the ability to manage global IP address pools. Global IP address pools permit a single pool of addresses to be shared by more than one network access server (NAS). This differs from local IP address pools which are defined on a per-NAS basis and which should not overlap.

radipad depends upon an Ascend vendor specific feature in the NAS. It does not work with non-Ascend equipment. This feature is the use of the Ascend-IP-Allocate (code 50) and Ascend-IP-Release (code 51) messages.

Ascend-IP-Allocate and Ascend-IP-Release messages are sent by the NAS after authentication has completed when the NAS needs to assign an address. The messages are sent to the RADIUS server defined on the NAS. When the RADIUS daemon receives the message, it generates a request to radipad to take the appropriate action. radipad then sends a message back to the RADIUS daemon, which generates a reply to the NAS.

A user is configured to use a global IP address when the user entry contains pool assignment of 65535 (Ascend-Assign-IP-Pool = 65535). This tells the NAS that it should request an IP address at assignment time. The user entry must also contain the name of the pool from which radipad should draw the address (Ascend-Assign-IP-Global-Pool = "global-pool-zzz").

In addition to modifying the individual user entry, two pseudo-user entries must be added to the users file. The first specifies the address pool information. This name of the pseudo-user corresponds to the name of the Ascend-Assign-IP-Global-Pool. The second pseudo-user entry specifies the address of the host running the radipad server. The name of the pseudo-user is "radipa-hosts".

The radipad server is normally invoked at boot time at the same time as the radiusd server. It will not run as a services on Windows/NT installations. Radipad will then listen for requests to connect to the radipad port (default 9992/tcp).

The radipad port may be changed when radipad is started by defining the radipad port in the system services file or database. The same service port must be defined on any system that will be using a NavisRadius server that communicates with radipad.

Options

Following are the options used with `radipad.`

Table A-6.   Table describes the options to use with radipad

Option	Description
`-d <sec>`	Wait delay seconds before answering a request. This debug option is not intended for normal operation.
`-h`	Causes a usage (help) message to be placed on the terminal screen.
`-n`	The radipad server does not recover the state information at startup from radipad.state. It should be used if the server has been down for an extended period of time.
`-p`	Report diagnostics to the screen rather than the logfile.
`-v`	Causes the radipad server to place its version information on the terminal screen.
-x	Allows the operator to turn on debugging output messages.

Commands to reset the log file

Radipad also creates a log file that can be updated as follows:

Table A-7. Table describes commands to update the log file

System	Command	Description
Windows/NT	Control-break	Resets logfile (backup the current logfile and open a new one).
	Control-C	Dumps state in the logfile
UNIX	SIGUSR1	Resets logfile (backup the current logfile and open a new one).
	SIGUSR2	Dumps state in the logfile

Exit Status

Normal successful completion returns zero to the system.

See also

radiusd, services, users

radiusd

Name

`radiusd` - Remote Authentication Dial In User service daemon

Synopsis

```
radiusd [ -d raddb_dir] [ -a acct_dir] [ -c cwd] [ -C]
[ -g ‘logfile‘ / ‘syslog’ / ‘stderr’ ] [ -h]
[ -p auth_port] [ -q acct_port] [ -f fsm]
[ -l format] [ -pp auth_relay] [ -qq acct_relay]
[ -r primary realm separator]
[ -rrsecondary_realm separator] [-t timeout]
[ -T timeout] [s] [ -Sa ] [-Sf] [-x ] [ -v] [ -z ]
[ -i] [ -P ] [ -oa]
```

Description

`radiusd` handles Access-Requests for user authentication from Ascend NavisRadius clients. These clients can be Network Access Servers (NAS) or other RADIUS servers. Authentication requests come to `radiusd` in the form of UDP packets conforming to the Ascend NavisRadius protocol.

`radiusd` collects authentication requests and processes them depending on their type. “Description” on page A-12 describes the types of authentication requests. If requested, `radiusd` can authenticate a user by calling upon other RADIUS servers, authentication services such as Kerberos, and operating system services, such as the UNIX system subroutines which access the `/etc/passwd` file.

When `radiusd` receives an authentication request from a client, in the form an Access-Request packet, it validates the client. Then `radiusd` consults a local database of users in a `users` file to find a user name that matches the one in the request, (see “users” on page A-42). The matching user entry in the `users` file contains a list of requirements which must be met before `radiusd` authenticates the user. These requirements are called check-items. Usually, one of the check-items is the user's password. For example, if the user's profile in the `users` file indicates the user's password is a check-item, the password received by `radiusd` must match the value of a user password attribute in the `users` file or the user will not be authenticated. There is no set list of requirements for all users and each may be entirely different, although some users with common connection needs may be nearly identical.

If any condition of the requirements is not met, `radiusd` sends an Access-Reject response to the client that requested authentication. If all the conditions are met, `radiusd` places a list of configuration values called reply-items in an Access-Accept packet and sends it to the client.These reply-items often include the type of service the user is allowed to use and other values necessary to deliver that service, such as the link protocol.

Ascend NavisRadius files

A number of files are installed with Ascend NavisRadius and used by `radiusd`. The `authfile`, `clients`, `dictionary`, and optionally, the `users` files, are read into resident memory tables.

The RADIUS server is refreshed whenever `radiusd` receives a HUP signal or the service is refreshed. Table A-8 lists other signals `radiusd` can receive and their effect.

Table A-8. Signals the RADIUS server can receive.

Signal	**Effect**
INT	Initializes only the AATV modules.
TERM	Provides an orderly way of shutting down the RADIUS server.
USR1	Turns on debugging, much as the `-x` option does, except that repeating the USR1 signal increases the debugging level. (See below for more information about the `-x` flag.)
USR2	Turns off debugging.

Ascend NavisRadius Installation

Ascend NavisRadius Ports

The following lines can be added to a system's "services" file when Ascend NavisRadius is installed. The information indicates that `radiusd` listens for Access-Requests on port 1812 and listens for Accounting packets on port 1813.

```
radius  1812/udp
radiusacct  1813/udp
```

Running as other than root

Note that the Ascend NavisRadius server need not run as root (UNIX) or Administrator (NT), although it normally does. It may be safer to pick a less powerful user who has no password and is used only for administrative purposes, unless the server needs super user privilege to access a shadow password file. Options

Table A-9 lists the `radiusd` options and their effect on the RADIUS server.

Table A-9. `radiusd` *options*

Option	**Description**
`-a` acct_dir	Allows the user to override the default accounting directory. Specify an alternate directory to contain the Ascend NavisRadius accounting detail files.
-C	Causes the RADIUS server to cache security token information for use in combination with the Ascend-Token-Expiry check-item attribute to enable with Ascend Pipeline CACHE-TOKEN authentication.
`-c` cwd	Allows the user to override the current working directory, or the default. Specify an alternate directory name. In UNIX this option only affects file system operation for files specified with relative file names that don't contain a leading forward slash (/) character.

Table A-9. `radiusd` *options*

Option	**Description**
`-d` database_directory	Allows the user to override the default database and configuration file directory. Specify an alternate directory name to house the Ascend NavisRadius `authfile`, `clients, dictionary,` and `users` files. See `authfile, clients, dictionary,` and `users` for more information.
`-f` fsm_file	Allows user to specify an alternate *finite state machine* table instead of the default `*.fsm` file. This option is supported by the Ascend NavisRadius software, but Ascend does not document the creation of alternative *finite state machine* tables.
`-g` ‘syslog’ | ‘logfile’ | ‘stderr’	Allows the user to specify whether to use `syslog(3)` style, logfile style or stderr logging for warning, error and informational messages. It is possible to specify daily or weekly renaming. You can also specify which weekday starts the week. Archiving, with compression of the Ascend NavisRadius logfile, is also supported. See the RADIUS_COMPRESS environment variable.
`-h`	Causes the Ascend NavisRadius server to place a usage (help) message into stdout.
`-i`	Enables radiusd to perform iPass authentication and accounting.

Table A-9. `radiusd` *options*

Option	Description
-l logfile format	By default radiusd automatically rotates the logfile in the *.../raddb* directory. The logfile is rotated to a file with an extension representing the date. The default extension is ".%y%m%d" (for example, yymmdd). The decision on when to rotate the file is also based on the file name extension chosen. When the change in date would cause the extension to change, the current logfile is closed and a new file is opened. The default is to rotate the file each day. You may modify the extension and thus, the frequency of the automatic logfile rotation, by supplying a string in strftime(3) format with the -l option. For example, -l ".%W" will rotate the file once a week. To disable automatic file rotation you may specify. -l "". Old logfiles are also automatically compressed using the program specified with the RADIUS_COMPRESS environment variable.
-oa	Causes the RADIUS server to look for the RADACCT user and to store accounting information using ODBC into a database table. .
-P	Causes the NavisRadius server to accept and process Password-Request packets to change the password for a user stored in a users file.
`-p` port	Allows user to specify an alternate authentication port number instead of the default port 1812.
`-pp` port	Allows user to specify an alternate authentication relay port number instead of the default port 1812.
`-q` acct_port	Allows user to specify an alternate accounting port number instead of the default port 1813.
`-qq` acct_port	Allows user to specify an alternate accounting relay port number instead of the default port 1813.

Table A-9. `radiusd` *options*

Option	**Description**
`-r` realm_separator	Allows user to specify the character the server recognizes as separator between a user name and a realm name.
`-rr secondary` realm_separator	Allows the user to specify an alternate secondary realm separator. The default secondary realm separator is '/'. When the Authentication-Type is realm, the username is searched for the primary realm separator. If the primary realm separator is not found, the username will searched for the secondary realm separator and if found, the username will be split into "realm / user".
`-s`	Places the Ascend NavisRadius server into the single process (non-spawning) mode, versus the multi-threated process mode. This option should only be used when debugging certain problems, as it causes a major decrease in performance.
`-Sa`	Causes the NavisRadius server to allow authentication to proceed as if the user limit had not been reached if no response is received from radstated, the RADIUS state daemon, when a user has a Simultaneous-Use limit.
`-Sf`	Causes the NavisRadius server to fail an authentication attempt, as if the user limit had been reached if no response is received from radstated, the RADIUS state daemon, when a user has a Simultaneous-Use limit.
`-t` timeout	Allows the user to specify a timeout value for the `select` system call which is different from the default timeout value of fifteen minutes. Giving the `-t` option a value of zero (`-t0`) puts the server into a blocking mode. `radiusd` never times out and terminates, but waits at the `select` call forever.
-T timeout	Allows the administrator to specify a minimum timeout period (in seconds) for user prompt responses different from the default 60 seconds.

Table A-9. `radiusd` *options*

Option	Description
`-u`	Specifies to NOT read the `users` file into the internal data structures. This function is supported; however, it is generally not used.
`-v`	Causes the Ascend NavisRadius server to place its version information on the terminal window.
`-x`	Allows the user to turn on the debugging output. The `-x` option can be repeated on the command line to increase the level of debugging information you receive. • `-x` provides minimal debugging output, send/receive a/v pairs, etc. • `-x-x` provides `-x` level debugging and *finite state machine* high level output and some function tracing • `-x-x-x` provides `-x-x` level debugging and the remaining function tracing • `-x-x-x-x` provides`-x-x-x` level debugging and *finite state machine* low level output and low level config files Note: Debugging output is directed to the `radius.debug` file. The `-x` option turns off some of the daemon behavior of the server, such as disconnecting from the controlling terminal. The maximum debug level is 4 (`-x-x-x-x`).
`-z`	Causes the Ascend NavisRadius `logfile` and `debug` file to be emptied, but only if the debugging option `-x` is enabled. This option has no effect on the `logfile` if the `-g` option specifies `syslog`(3) style logging
`-noservice`	Do not run as service (on NT only).

Exit Status

Table A-10 lists the `radiusd` exit status codes and associations. You can also find information about error termination conditions in `logfile` or `syslog` entries, depending upon the server's configuration.

Table A-10. `radiusd` *exit status values.*

Codes	**Association**
255 (-1)	dict_init
254 (-2)	config_init
253 (-3)	init_fsm
252 (-4)	config_files
251 (-5)	disconnect
250 (-6)	open PID file
249 (-7)	SIG_FATAL
248 (-8)	usage
247 (-9)	user_update
246 (-10)	version
245 (-11)	setupsock (can't bind, Ascend NavisRadius already running?)
244 (-12)	init_id_to_key
243 (-13)	list_copy
242 (-14)	find_state
241 (-15)	chdir
240 (-16)	hostname

See also

authfile, clients, dictionary, users, radcheck, radpwtst, radcheck

radstated

Name

The RADIUS State Daemon for tracking user connections

Synopsis

radstated [-d directory] [-g ’logfile’ | ‘stderr’] [-l format]

[-h] [-v] [-n] [-x] [-z] [-A] [-C community]

Description

The RADIUS State Daemon (radstated) tracks connections using the RADIUS accounting information provided by a Network Access Server (NAS). This state information is then used to set limits on the number of times a user or group of users may connect across one or more NASes.

The state information is checked when a user or group limit exists for a user. A user is configured to use a user session limit when the user entry includes the Session-Limit attribute as a check-item. The Session-Limit is an integer which defines the maximum number of ports that the user is permitted to use. A user is configured to use a group session limit when the user entry includes the Group-Session-Limit and Group-Limit-Name attributes as check-items. The Group-Session-Limit defines the maximum number of ports that may be in use by the group when the user tries to connect. The Group-Limit-Name defines the group to which the user belongs. Different users who share the same Group-Limit-Name may have different values for Group-Session-Limit. A user may have both user and group session limits defined as check-items.

If the user session limit (Session-Limit) is reached, radstated attempts to verify the state information it has accumulated. radstated checks each port the user is believed to be using SNMP to verify that the port is still in use by the user. This verification uses parts of the Ascend vendor MIB and will therefore not work with another vendor NAS. This default verification behavior may be disabled with the -A command line option. radstated does not attempt to verify the state information when the group session limit is reached.

In order for radiusd to find the radstated server, the address of the host running radstated must be defined using a pseudo-user entry in the users file. The name of the pseudo-user is "radstate-hosts".

The radstated server is normally invoked at boot time at the same time as the radiusd server. radstated will then listen for requests to connect to the radstated port (default 9993/tcp) and radstate-acct (default 9994/udp) port.

The radstate-service or radstate-acct ports may be changed when radstated is started by defining the "radstate-service" or "radstate-acct" port in the system services file or database. The same service port(s) must be defined on any system that will be using a NavisRadius server that communicates with radstated.

Note: To have `radstated` update its state file and print its state in its logfile, use

- Control-C on Windows NT
- SIGHUP on UNIX

Options

The `radstated` options are described below.

Table A-11. Displays and describes options to use with `radstated`

Option	Description
`-A`	This option will cause radstated to use only the RADIUS Accounting-Request messages and does not try to verify user session limits using SNMP. radstated uses Accounting-Request messages to track the start and stop of sessions. When radstated is used to limit user sessions with the Simultaneous-Use attribute in the radiusd users file, radstated will try to verify the connection count using SNMP if the session limit is reached. If radstated is used in a mixed-vendor environment, radstated will be unable to use SNMP to verify the connection count because the SNMP objects are in a vendor specific MIB.
`-C` community	SNMP community: string name to be used in SNMP requests to verify the number of a user's simultaneous connections.
`-d` directory	Allows the user to override the default database and configuration file directory by specifying an alternate directory name containing the RADIUS IR authfile, clients, dictionary, radius.fsm and users files instead of the default /etc/raddb directory. See authfile, clients, dictionary, radius.fsm and users for more information.
`-g` 'logfile'|'stderr'	Allows you to specify whether to use logfile style or stderr logging for warning, error and informational messages.The default is logfile style.

Table A-11. Displays and describes options to use with `radstated`

Option	Description
`-h`	Causes a usage (help) message to be placed in stout.
`-l` format	By default radiusd automatically rotates the logfile in the .../raddb directory. The logfile is rotated to a file with an extension representing the date. The default extension is ".%y%m%d" (e.g. yymmdd). The decision on when to rotate the file is also based upon the file name extension chosen. When the change in date would cause the extension to change, the current logfile is closed and a new file is opened. The default is to rotate the file each day. You may modify the extension and thus the frequency of the automatic logfile -l option. For example, -l ".%W" rotates the file once a week. To disable automatic file rotation you may specify -l "". Old logfiles are also automatically compressed using the program specified with the RADIUS_COMPRESS environment variable.
`-n`	radstated does not recover the state information at startup from radstated.state.
-v	Causes the radstated server to place its version information onto stdout and then exit

Table A-11. Displays and describes options to use with `radstated`

Option	Description
-x	Allows the user to turn on debugging output: -x minimal debugging output -x -x information and function tracing. -x -x -x above -x -x -x -x more packet information, more function tracing Debugging output is directed to the adstate.debug file. The -x option causes the daemon to run in single process mode. radstated will not fork and detach from the controlling terminal.
`-z`	Causes the radstated logfile and debug file to be emptied, but only if -x is enabled. This option has no effect on the logfile if the -g option specifies syslog style logging.

See also

clients, dictionary, radiusd, services, signal, syslog, users

radpwtst

Name

`radpwtst` - authenticates a user’s password

Synopsis

```
radpwtst [ -c code] [ -d directory] [ -f file]
[ -g dummy-profile-name ] [ -h] [ -i NAS name/ip_address ]
[ -l aysnc_port ] [ -p UDP_port ] [ -r retries ]
[ -s server ] [ -t timeout ] [ -n ] [ -u type ]
[ -v version ][ -w password] [ -x]
[ -:<attribute>=<attr-value>] username
```

Description

`radpwtst` uses a Ascend NavisRadius server to authenticate a user. Following is the process when the userid of the user being authenticated has been entered on the command line the authentication:

- `radpwtst` prompts for the password that matches the userid
- `radpwtst` forwards the userid/password tuple to a Ascend NavisRadius server

An exact match is required for successful authentication.

userid@realm format

When the userid is given in the format `user@realm`, it is assumed that the user is a member of an authentication realm listed in the Ascend NavisRadius server’s `authfile` file. This file is searched for a matching userid/password combination. `authfile` is further assumed to be in the Ascend NavisRadius default directory. See “authfile” on page A-2 for more information.

userid format

When the optional *`@realm`* is omitted from the userid, the userid is sought in the Ascend NavisRadius server's `users` file.

DEFAULT user

In either case, *userid@realm* format or userid format, an exact match of the userid/password tuple is required. If the match fails, the Ascend NavisRadius server checks the appropriate file for a DEFAULT entry. The DEFAULT entry should contain information about how to authenticate the user.

`radpwtst` displays one of two messages on standard output, depending on the success of the authentication:

```
authentication OK
```

```
userid authentication failed
```

Using radpwtst to free allocated Addresses

To free address x.x.x.x allocated by radipad running on machine y.y.y.y, perform the following steps.

1 Start radiusd on y.y.y.y

2 Run radpwtst on y.y.y.y as follows:

```
radpwtst -i n.n.n.n -c51 -:Framed-IP-Address=x.x.x.x -r
-w 0 -x -x -w <password> <username>
```

To free all the addresses allocated by a NAS n.n.n.n, perform the following steps:

1 Start radiusd on y.y.y.y

2 run radpwtst on y.y.y.y as follows

```
radpwtst -i n.n.n.n -c51 -:Framed-IP-
Address=255.255.255.255 -r 0 -x -x -w
<password> <username>
```

Options

Table A-12 lists the `radpwtst` options and their descriptions.

Table A-12. `radpwtst` options and their descriptions

Option	Description
`-c` code	Allows user to specify several Ascend NavisRadius packet type codes. Codes can be one of: • 1 - (Access-Request) • 2 - (Access-Reject) • 3 - (Access-Accept) • 4 - (Accounting-Request) • 5 - (Accounting-Response) • 7 - (Password Change Request) • 8 - (Password Change ACK) • 9 - (Password Change NAK) • 11- (Access Challenge) • 12 -(Status-Server) • 50 -(IP Address Allocate) • 51 -(IP Address Release)
`-d` directory	Allows the user to specify an alternate directory for the Ascend NavisRadius `authfile`, `users`, `clients` files. This replaces the default directory which is `/etc/raddb`. An error is displayed on stdout if the Ascend NavisRadius configuration files are not found. **Note:** If the machine running `radpwtst` is different than the machine running the Ascend NavisRadius server, make sure that the contents of one machine's configuration files are identical to the contents of the other machine's configuration files.

Table A-12. `radpwtst` options and their descriptions

Option	Description
`-f` file	Allows the user to specify a prefix for a file in the `users` file format. See“users” on page A-42. The file name is assumed to be `prefix.users` and it is assumed to be in the Ascend NavisRadius configuration file directory, (see `-d` above). This file contains arbitrary check-items and reply-items (attribute and value pairs) for pseudo-users whose names may be specified by the `-g` option. If no `-g` option is given, the DEFAULT entry is used if it is present. This is the way arbitrary attribute/value pairs are communicated to remote Ascend NavisRadius servers. See “users” for more information.
`-g` dummy-profile-name	Allows the user to specify an arbitrary pseudo-user named group in the file named by the `-f` option. See -f option for more about sending arbitrary attribute/value pairs to remote Ascend NavisRadius servers.
`-h`	Causes a usage (help) message to be placed in stout.
`-i` NAS Name or IP Address	Allows the user to specify the name or IP address of the NAS.
`-l` async_port	Allows the user to specify a an async port number that’s different than the default async port of 1.
-n	Allows the user to force the Authentication-only value to be used in the attribute-value pair Service-Type.
`-p` UDP_port	Allows the user to specify a UDP port on the server that’s different than the default, 1812.

Table A-12. `radpwtst` *options and their descriptions*

Option	Description
`-r` retries	Allows the user to specify that the maximum number of retries be something other than the default 10 retries
`-s` server	Allows the user to specify the name or IP address of the RADIUS server.
`-t` timeout	Allows the user to specify an alternate time-out value other than the default 3 seconds. Timeout values are x seconds.
`-u` type	Allows the user to specify one of the following Service-Type values instead of the default auth value: • `admin, arades, auth, challenge, chap, ra dumb, exec, outbound, ppp, slip, dbadmin, dbdumb, dbppp, dbslip` (`db` stands for dial back.) **Note:** The default auth fails if the Access-Request produced by `radpwtst` contains no password or an empty password (`default` or `-c1`). When the Service-Type is Authenticate-Only, the Ascend NavisRadius server requires a valid, non-empty password in Access-Request packets.
`-v`	Prints the radius to be used to build the program. If the option is -v1 or -v2 the request is built according to version 1 or 2 of the radius protocol, respectively.
`-w` password	Allows the user to provide a password on the command line. The user is not prompted for a password.
`-x`	Allows the user to turn on the debugging output

Table A-12. `radpwtst` options and their descriptions

Option	Description
-:<attribute>=<value>	Text following the colon (:) character specifies the value of any attribute in the `dictionary`. The syntax is identical to that of reply-items, as described in “users” on page A-42.

Exit Status

Table A-13 lists the exit status values `radpwtst` returns to the system.

Table A-13. radpwtst exit status values.

Value	Description
1	timeout error
0	successful completion
-1	local error
-2	Ascend NavisRadius server error

See also

authfile, client, dictionary, radcheck, radiusd, **radcheck, users**

users

Name

`users` - Ascend NavisRadius user security and configuration file

Synopsis

UNIX

`/etc/raddb/users`

Windows NT

`c:\Ascend\NavisRadius\raddb\users`

Description

The `users` file is installed in the Ascend NavisRadius server's default directory, unless configured differently by the system administrator. `users` is read by `radiusd` at startup and whenever `radiusd` receives a `HUP` signal (UNIX) or the server is refreshed (Windows NT).

`users` contains a list of the users the Ascend NavisRadius server can authenticate for clients and other servers. Ascend NavisRadius clients are Network Access Servers (NAS), like routers, or other Ascend NavisRadius servers.

`users` may contain comments, which are indicated by leading pound sign (#) character. Comments and blank lines inserted in `users` are ignored.

Commas may be used to separate items in any line, or at the end of any line. Ascend NavisRadius treats commas as though they are whitespace.

Initial Lines

`users` may contain one or more lines of information per Ascend NavisRadius user entry. The first line of each user entry consists of one or more fields, in the sequence shown below. The fields are separated by whitespace.

```
<user-name> <check-item> [, <check-item>]...
```

Example:

```
george Password=”casablanca”
```

Special user names

Ascend NavisRadius recognizes four special user names which are used for these reasons:

- the default for all user names which do not match any previous entries
- to hold non-framed reply-items
- to hold PPP reply-items
- to hold SLIP reply-items

The special user names include, DEFAULT which specifies how to authenticate user names which do not match any previously parsed entries. DEFAULT should be the last entry in the `users` file.

The special user name ODBCACCT specifies that accounting information will be sent to the data source indicated by the value of the profile’s Password attribute.

The other special user names `dumbuser, pppuser` and `slipuser` allow a user seeking authentication to submit the same account name for any of the framed or non-framed protocols.

Additional lines

The initial line of a user entry may be followed by additional lines containing reply-items, or attribute/value pairs, that the Ascend NavisRadius server sends back to the requesting Network Access Server (NAS) or another Ascend NavisRadius server. The additional lines must begin with white space. The reply-items may include PPP configuration values, the name of the host to which the user wishes to connect, or any other appropriate attribute/value translations listed in the `dictionary` file.

Additional lines in a `users` file entry have this syntax.

```
whitespace reply-item,
```

Example:

```
Framed-Protocol=PPP
```

Authentication-Type attribute and ODBC support

The Authentication-Type attribute provides the link between the `users` file's user profiles and authentication and accounting attributes stored in the table(s) of a Database Management System (DBMS) that complies with the Open Database Connectivity (ODBC) standard. The support of ODBC tables brings the power and speed of relational database computing to Ascend NavisRadius authentication, authorization and accounting.

The initial line of the user profile that provides a connection to the table has a unique format. The value of the Password attribute include four entries that enable Ascend NavisRadius to access the table's information. The format of the initial line is:

```
<username> Authentication-Type = “<database source
name>:database table name>:<table access name>/<table access
password>
```

Example:

```
jsmith@ascend  Authentication-Type = “qesyb:atable:
aname/apasswd”
```

The user name may be that of a single user such as `jsmith@ascend`, a realm, `ascend.com` or DEFAULT.

Table A-14. Required Password value for support of ODBC tables

Value	**Description**
ODBC	Flag telling Ascend NavisRadius to search for a related database table
database source name	Name of database server the Ascend NavisRadius ODBC Manager consults
database table name	Name of database table • may be one of several on server • may be associated with realm name • Example: ascendtabl for ascend.com realm
table access name	Name provided by Ascend NavisRadius server when accessing a database table • different for each table
table access password	Password provided by Ascend NavisRadius server when accessing a database table • different for each table

Vendor-specific attributes

Vendor specific attributes may be used in place of normal check-item and reply-item attributes in the user entry. Vendor specific attributes have the form `<vendor>:<attribute>.`

Example:

```
Ascend:Ascend-Metric
```

Other users files

Although `users` is installed with Ascend NavisRadius and is the first file consulted by Ascend NavisRadius when it receives an Access-Request, there may be other `users` files which are related to separate realms listed in the `authfile` file. These other `users` files share the naming convention `prefix.users`, where `prefix` is usually the name of the realm with which the users are associated.

Example:

Realm	**`Prefix.Users` file name**
ISP1	`ISP1.users`
admin	`admin.users`

Note that the users file may be converted into a dbm database by using the ‘buildbm’ utility. This file is maintained by the RADIUS administrator using a text editor.

See also

dictionary, radiusd, radcheck, radpwtst, dbm, signal

vendors

Name

`vendors` - Ascend NavisRadius file for mapping vendors to vendor codes

Synopsis

UNIX

`/etc/raddb/users`

Windows NT

`c:\Ascend\NavisRadius\raddb\users`

Description

The `vendors` file is installed in the Ascend NavisRadius server's default directory, unless configured differently by the system administrator. `vendors users` is read by `radiusd` at startup and whenever `radiusd` receives a `HUP` signal (UNIX) or the server is refreshed (Windows NT).

The `vendors` file is distributed with Ascend NavisRadius. You can use text editors such as vi or Notepad to edit the `vendors` file.

`vendors` contains a list zero or more vendor entries. Each vendor entry contains a vendor name and a vendor number. The vendor numbers are SMI Network Management Private Enterprise Code numbers as described in the RADIUS DRAFT RFC and RFC 1700. Each entry optionally contains an interim way of mapping attribute numbers assigned by vendors outside the RADIUS DRAFT conventions to the RADIUS vendor-specific attribute, which is defined in the RADIUS DRAFT as attribute number 26. This optional mapping is used on Ascend NavisRadius server inbound and outbound requests.

`vendors` may contain comments, which are indicated by leading pound sign (#) character. Comments and blank lines inserted in `vendors` are ignored.

The file contains a line of information for each vendor in the following form:

```
[<attribute-string> <value-string>] <vendor-code>
<vendor-name> [( [ <standard-value>
<vendor-specific-value>...])]
```

Example:

```
61 MERIT (211 211 213 213)
Ascend.attr Ascend.value 529 Ascend
(
172 172
156 156
)
```

Attribute-string and value-string options

The `attribute-string` and `value-string` are optional strings which default to `Attribute` and `Value` when not specified. Non-default strings may be used to specify vendor specific attributes and values in the `dictionary` file.

vendor-code field

The `vendor-code` field contains a number assigned to the vendor in the Assigned Numbers (IANA), RFC 1700.

Example:

Table A-15. Vendor Codes from RFC 1700

Vendor Name	Vendor Code
MERIT	61
Ascend	529
US Robotics	429

vendor-name field

The `<vendor-name>` field is the vendor name. The vendor name may appear in the `clients` file as a `type=vendor:nas` entry or in vendor-specific attribute names in the `dictionary` and `users` files.

standard-value and vendor-specific value options

The `standard-value` option is the external, or common, attribute number as seen in Ascend NavisRadius requests on the network.

The `vendor-specific-value` option is the internal attribute number, or the number the vendor assigned to the attribute when it was developed and added to RADIUS.

`standard-value` and `vendor-specific-value` optional fields may be repeated an optional number of times within the parentheses. These numbers are used to map attributes from the common attribute space defined in the RADIUS RFC to internal, non-conflicting vendor-specific attributes. This is necessary because some vendors assign vendor-specific attributes in the standard attribute space instead of in the vendor-specific attribute position defined in the RADIUS RFC.

See also

clients, dictionary, users

Debug, log and state files summary list

The following table lists the debug, log and state files for NavisRadius.

Table A-16. Table describes commands to update the log file

File	System	Description
Radiusd	Unix	logfile<log_format, mostly date> logfile (symbolic link to the current logfile. <log_format>) radius.debug
	Windows/NT	logfile.txt radius.debug.txt
Radstated	Unix	radstate.log<log_format,mostly date> radstate.debug radstate.ckp (state file)
	Windows/NT	radstate.log <log_format, mostly date>.txt radstate.debug radstate.ckp (state file)
Radipad	Unix and Windows/NT	radipad.logfile radipad.state

Radiusd.pid, radstate.pid, and radipad.pid files contain the version information and process ID of the respective processes.

All these files are present in the directory `/etc/raddb` on UNIX and `\Ascend\NavisRadius\raddb` or Windows/NT.

iPass notes

B

iPass background technical information

Introduction

This document provides background technical information about the iPass software. Its intended audience is the technical staff of an ISP.

Environment

The iPass system runs as a Unix daemon on hosts in the ISPs network.

The iPass Virtual NAS Daemon (vnas) is a ``virtual" terminal server or Network Access Server (NAS) which forwards authentication requests from iPass to a local authentication server such as RADIUS or TACACS. vnas may also authenticate via the unix password file, or a site specific authentication scheme.

This daemon communicates with the iPass Communications Daemon (ipcd) running at a central iPass site. Ipcd in turn, communicates with ipdd, the central data base daemon. All communication between iPass daemons is via the SSL protocol on standard sockets.

The default home directory of the iPass software distribution is /usr/ipass. This directory will be referred to in this manual as **$IPASS_HOME**.

Certificate and key management

Certificates are used within the iPass system to mutually authenticate both the client and server sides of an SSL connection.

The following authentication operations must take place:

- iprd must authenticate the ipcd process to which it has connected
- each ipcd process must authenticate the iprd process connecting to it
- ipcd must authenticate the vnas daemon process to which it has connected
- each vnas daemon process must authenticate ipcd processes connecting to it
- ipcd must authenticate the ipdd process to which it connects
- ipdd must authenticate the ipcd process connecting to it

Types of certificates

Each of the main iPass software components must have its own certificate. iPass Alliance Inc., is the Certificate Authority responsible for issuing these certificates.

Certificates are stored in the directory $IPASS/certs, where $IPASS is the iPass home directory (default is /usr/ipass). The certificate file for both iprd and vnas is named isp.pem.

Each system must also have a copy of the iPass Certificate Authority root certificate. It is stored in the same directory and is named ipassCA.pem.

Certificate creation

The CA root certificate is distributed as part of the iPass software.

The user certificates must be created during installation. This is performed   by the iPass installation and configuration utility program (ipass_config).

Key generation

A private key is generated by ipass_config during installation. The key is stored in $IPASS_HOME/keys/isp.pem and is DES encrypted. The key used to encrypt the private key is automatically generated based on several inputs including:

- the name of the program which is executing
- the isp_code specified during installation
- the hostname of the system on which the program is executing
- the IP address of the system on which the program is running

The intention is to automatically generate a unique key for each system, where the key cannot easily be guessed. The goal is to prevent the software from running on a different system without repeating the installation and configuration process.

iPass services

```
vnas 9999/tcp vnas #iPass Virtual NAS
```

The following is the port that the iPass central server listens on (this is for information only, don't add this to your services file)

```
# ipcd 9101/tcp
```

Installing the iPass software

Introduction

This document describes how to install or upgrade the iPass software on your UNIX-based system, how to perform initial testing and configuration, and how to configure the software for ``live" operation.

iPass recommends that the initial installation and testing of this software be done in a non-production environment. If you do not have a suitable test environment and must install directly on your production authentication servers, please take appropriate steps to ensure that you can revert to the previous system configuration if anything goes wrong.

If you are upgrading a system you should skip forward to the section titled, *“Upgrading an existing installation” on page B-7‘*

Installation

See Chapter 3, “Installing Ascend NavisRadius” for more information.

iPass software configuration

Table B-1. The subdirectories and their contents are listed below

Subdirectory	Contents
ipass	Contains a sample of the configuration file ipass.conf, and the ispform that needs to be filled out and sent back to iPass.
ipass/bin	Contains configuration scripts, authentication daemons, and the virtual NAS daemon.
ipass/certs	Contains the iPass Certificate Authority root certificate, and will contain the public certificate after it has been created.
ipass/lib	Contains object file libraries for your platform (required for custom installations).
ipass/keys	Contains the private key that will be created by the ipass_config script.
ipass/logs	Contains log and trace files.
ipass/test	Contains test utilities,

1 Prepare to configure the software for your local environment using the following commands:

```
cd /usr/ipass
bin/ipass_config
```

The ipass_config script generates a private key and stores it in the keys directory. It then asks a series of questions about your local environment. The information you supply generates a public key certificate request which is mailed to iPass. iPass will then generate the public certificate and will mail it back to you.

2 Place this certificate (i.e. the entire e-mail reply message) in the **/usr/ipass/certs** directory in a file named **``isp.pem''.**

(Do not over-write the file of the same name in the keys directory.)

Note: If you use something other than a Unix mail program to read mail, for example, if you use a Windows mail reader and then ftp the file over to the server, care must be taken to ensure the certificate is not corrupted.

After the file has been moved to the Unix server, make sure there are no extra line feeds at the end of each line. These will show up as ``^M" characters in the file.

3 ipass_config also stores the configuration in a file called “/usr/ipass/ipass.conf” file. This can be edited later if required.

Installing the vnas daemon

The vnas (Virtual NAS) daemon is located in the **/usr/ipass/bin** directory. The vnas daemon has support for authenticating with RADIUS, TACACS+, UNIX password, or a user defined method. The type of protocol vnas uses is determined by the vnas_protocol parameter in the ipass.conf file.

1 First you must add vnas to the Internet services file ”/etc/services” with a line as follows:

```
vnas      9999/tcp
```

2 The vnas daemon is typically started by inetd by modifying the inetd.conf file. To do this for most standard systems, the following line must be added to /etc/inetd.conf:

vnas stream tcp nowait root /usr/ipass/bin/vnas vnas

If you are using sinned, or some other modified INET connection manager, you will need to customize its configuration appropriately.

3 Tell the inetd daemon to re-read the inetd.conf file by sending it a SIGHUP signal. To do this, find out the process number of the inetd process by using the ``ps" command:

ps -ef | grep inetd (for Solaris or OSF/1)

4 Send the SIGHUP signal with the command ``kill -HUP xxxx".

``xxxx" is the process number returned by the ``ps' command.

Note: If the vnas daemon is authenticating against a RADIUS daemon, an extra step is required. If vnas is authenticating via a RADIUS server running on a different machine, the RADIUS server's clients file must be copied onto this machine or otherwise made accessible. In the case of a TACACS+ server running on a different machine, its config file must be copied to this machine. If vnas is running on the same machine as the RADIUS screen, a RADIUS clients file must be created, or an existing one modified.

These files contain the shared secret used between the NAS and the authenticating daemon. As the vnas daemon is acting like a NAS, a shared secret must be added to the RADIUS clients file or the name of the system that vnas is installed on. vnas uses the gethostname call to determine the name of the system it is running on.

Depending on the O/S, the gethostname call will either return the name in the format 'hostname.domain', or 'hostname'.

5 To find out the name format to enter in the clients file, type the command 'hostname'.

Upgrading an existing installation

If you are installing a new release of the iPass software on a system which is already running an older release, you may use the following procedure.

As always we recommend doing the upgrade in a test environment before modifying your production server(s). This is particularly important if you have to first integrate iPass changes into a customized authentication daemon, or are using a site-specific authentication or accounting method in the vnas daemon.

Before doing anything, you should first shut down any iPass related daemons. Ideally, you should do the upgrade while in single-user mode. If this is not possible, you must at least shut down the NAS authentication daemon(s) and disable the vnas daemon in inetd.conf.

Once everything related to the iPass software is stopped, unpack the distribution following the instructions described in the section ``Installation Steps'' at the beginning of this document.

Do not run the configuration script (bin/ipass_config) again. You can reuse the existing keys and certificates from your previous installation. They will not be disturbed by the upgrade procedure. If you are not running in single user mode, you need to remember to send a SIGHUP signal to the inetd process to start the new vnas service. Otherwise, if you are running in single user mode, you may return to multi-user mode.

Testing the installation

After the software has been installed, the isp.pem certificate installed, and the ipass.conf file edited, the next step is to test the operational status of the iPass software in the current configuration.

Testing the vnas daemon

The first stage in the test process is to verify that the vnas daemon is installed and configured correctly.

Included in the /usr/ipass/test directory is a utility called 'check-vnas' that may be used for this purpose. The syntax for this utility is 'check-vnas username'. Use a local account for the user name and supply the password for the local account when prompted to do so.

The check-vnas utility will contact the local vnas daemon in the same manner that the central iPass server would.   The vnas daemon will then contact the local authentication server to authenticate the user. Diagnostic output showing the transactions and any error messages is displayed on stdout.

Sample trace output is shown below with the time stamp information removed for clarity.

```
iprd[23278]: authcache_init: authcache initialized, size 500

Password:

VNAS server will be authenticating using radius

Contacting vnas server at 10.0.0.16:9999 to authenticate
user fred

iprd[23278]: net_ip_connect: s= 8

iprd[23278]: connect to <10.0.0.16> on port 9999 ...

iprd[23278]: ipass_verify: appl cert, isp_code=<12345>,
ip_addr=<10.0.0.16>

iprd[23278]: ipass_verify: root cert OK

iprd[23278]: net_ip_connect: SSL setup took 1075145 micro-
seconds

Connected to vnas, sending request ...
```

```
VNAS return status = accept
Service Type:            framed_user
Framed Protocol:         PPP
```

If instead of a return status of ``accept," you see the message ``Error getting reply," your vnas daemon encountered problems. You can determine the exact nature of the problem by examining syslog output, and alternately by enabling debug tracing for vnas.

Note that if the vnas protocol is set to "RADIUS" and there's a problem contacting the radiusd, vnas tries 5 times at 30 second intervals before giving up and you see the ``radius_authenticate: connection timed out" message.

Testing remote authentication

After the vnas daemon has been tested and any problems fixed, the next step is to send a transaction to the central iPass server.

The check-ipgen utility eliminates the need for a NAS for the initial test confirming the operational status of the iPass software, and providing voluminous debug output for trouble-shooting.   It sends transactions through to iPass to simulate a complete session by a roaming user. You may want to run it in such a way so that you can review many screens full of information after it finishes - perhaps in an xterm with a large scroll-back buffer.

The general usage of this utility is 'check-ipgen username@ipass.authdom'. For the loop-back test, use a local account for the user name, and ``ipass.test" for the iPass authentication domain name. Supply the password for the local account when prompted to do so. The sequence of events for a loop-back test is as described below. Relevant trace output for each step is shown.

- The check-ipgen process acts as a local iPass daemon by forwarding the request to the central iPass server. Before forwarding the request, it replaces the authentication domain name in the request with the value configured as the local authentication domain.

```
iprd[23334]: ipass_remote_auth:
    iprd[23334]: NAME = auth_request
    iprd[23334]: ver = <3.0>
```

```
iprd[23334]: isp_code = <002>
iprd[23334]: auth_domain = <seachange.com>
iprd[23334]: userid = <fred>
iprd[23334]: local_time = <Wed Feb 19 12:09:49
EST   1997>
iprd[23334]: nas_ip = <192.139.22.3>
iprd[23334]: session_id = <2222222222>
iprd[23334]: nas_port = <50>
iprd[23334]: event_tmstmp = <856372189>
iprd[23334]: direction = <unknown>
iprd[23334]: service_type = <framed_user>
iprd[23334]: framed_protocol = <PPP>
iprd[23334]: source_ip = <10.0.0.16>
iprd[23334]: passwd is 4 chars long
iprd[23334]: net_open: calling
net_connect(auth1.ipass.com, 9101, 1)
iprd[23334]: ipass_verify: appl cert,
isp_code=<Engineering>, ip_addr=<(null)>
iprd[23334]: ipass_verify: root cert OK
iprd[23334]: net_connect: SSL setup took 1892986
microseconds
iprd[23334]: net_open:Connected to iPass server
auth1.ipass.com on port 9101
iprd[23334]: ipass_remote_auth: done sending
transaction... getting reply...
```

- The central iPass server forwards the request to your local vnas daemon.
- The local vnas daemon authenticates 'username' by contacting the local authentication daemon. The authentication domain has been stripped off to ensure that the local authentication daemon does not recognize this request, as an iPass roaming user and loop again through the iPass central server.
- The local vnas receives the authorization status from your local authentication daemon and returns it to the central iPass server.

- The central iPass server returns the authorization status to the check-ipgen program.

```
iprd[23334]: post_get_status:
     iprd[23334]:   NAME = auth_reply
     iprd[23334]:   ver = <3.0>
     iprd[23334]:   status = <accept>
     iprd[23334]:   time_to_live = <10000>
     iprd[23334]:   session_limit = <10000>
     iprd[23334]:   service_type = <framed_user>
     iprd[23334]:   framed_protocol = <PPP>
     iprd[23334]:   auth_domain = <seachange.com>
     iprd[23334]:   userid = <fred>
iprd[23334]:   nas_ip = <192.139.22.3>
     iprd[23334]:   net_source_ip = <192.139.22.2>
     iprd[23334]:   isp_code = <002>
     iprd[23334]:   session_id = <2222222222>
     iprd[23334]:   local_time = <Wed Feb 19 12:09:49
     EST 1997>
     iprd[23334]:   tr_seqno = <db1:21326>
     iprd[23334]:   ipass_remote_auth: status = accept.
     iprd[23334]: authcacheadd: adding fred@seachange.com,
     ttl = 10000
     ./check-ipgen: time to authenticate: 17 seconds.
     ./check-ipgen: ipass_remote_auth(fred@ipass.test) = 1,
     '<no message>'
```

- The check-ipgen program verifies that a successful authentication was cached locally by re-authenticating the user. The time to return the second authorization should be negligible.

```
./check-ipgen: testing cache for fred@ipass.test.
     iprd[23334][23336]: authcachefind: cached entry
      fred@seachange.com valid for another 9999 seconds.
```

```
    iprd[23334][23336]: ipass_remote_auth: using cached
    response for fred@clapton.net

    ./check-ipgen: ipass_remote_auth(fred@ipass.test) = 1,
    '<no message>'

    ./check-ipgen: time to authenticate from cache: 0
    seconds.
```

- The check-ipgen program sends an accounting session start transaction to the central iPass server.

```
iprd[23334]: post_acct_request:

    iprd[23334]:   NAME = acct_request

    iprd[23334]:   isp_code = <002>

    iprd[23334]:   auth_domain = <seachange.com>

    iprd[23334]:   userid = <fred>

    iprd[23334]:   local_time = <Wed Feb 19 12:10:03
    EST 1997>

    iprd[23334]:   nas_ip = <192.139.22.3>

    iprd[23334]:   session_id = <085637220301>

    iprd[23334]:   nas_port = <50>

    iprd[23334]:   event_tmstmp = <856372186>

    iprd[23334]:   direction = <unknown>

    iprd[23334]:   service_type = <framed_user>

    iprd[23334]:   framed_protocol = <PPP>

    iprd[23334]:   source_ip = <10.0.0.16>

    iprd[23334]:   session_time = <0>

    iprd[23334]:   user_ip = <10.0.0.101>

    iprd[23334]:   user_mask = <255.255.255.255>

    iprd[23334]:   acct_type = <START>

    iprd[23334]:   net_open: calling
    net_connect(auth1.ipass.com, 9101, 1)

    iprd[23334]:   ipass_verify: appl cert,
    isp_code=<Engineering>, ip_addr=<(null)>

    iprd[23334]:  ipass_verify: root cert OK
```

```
iprd[23334]:  net_connect: SSL setup took 3017159
microseconds

iprd[23334]:  net_open: Connected to iPass server
auth1.ipass.com on port 9101

iprd[23334]:  post_acct_request: done sending
transaction...  getting reply...
```

The central iPass server logs the session start in the iPass database.

- The central iPass server forwards the accounting session start transaction to the local vnas daemon.
- The local vnas daemon forwards the accounting session start data to the local authentication server for local logging.
- The local authentication server logs the session start data and returns an acknowledgment of this logging to the local vnas daemon.
- The local vnas daemon returns the accounting acknowledgment to the central iPass server.
- The central iPass server returns the accounting acknowledgment to the check-ipgen program.

```
    iprd[23334]:   post_get_status:
    iprd[23334]:   NAME = acct_reply
    iprd[23334]:   tr_seqno = <db1:21327>
    iprd[23334]:   tr_gmt_time = <19-Feb-1997 16:56:55>
    iprd[23334]:   billing_code = <B:9042>
    iprd[23334]:   vnas_list = <192.139.22.2>
    iprd[23334]:   status = <accept>
    iprd[23334]:   post_acct_request: status = accept.
    ./check-ipgen: ipass_remote_acct(fred@ipass.test) = 0,
''
```

- Finally, the check-ipgen program sends a simulated accounting session end transaction and receives the response.

```
iprd[23334]:   post_acct_request:
iprd[23334]:   NAME = acct_request
iprd[23334]:   isp_code = <002>
```

```
     iprd[23334]:   auth_domain = <seachange.com>
     iprd[23334]:   userid = <fred>
     iprd[23334]:  local_time = <Wed Feb 19 12:10:12
     EST 1997>
     iprd[23334]:   nas_ip = <192.139.22.3>
     iprd[23334]:   session_id = <085637220301>
     iprd[23334]:   nas_port = <50>
     iprd[23334]:   event_tmstmp = <856372212>
     iprd[23334]:   direction = <unknown>
     iprd[23334]:   service_type = <framed_user>
     iprd[23334]:   framed_protocol = <PPP>
     iprd[23334]:   source_ip = <10.0.0.16>
     iprd[23334]:   input_packets = <18546>
     iprd[23334]:   output_packets = <7274>
     iprd[23334]:   input_chars = <2373888>
     iprd[23334]:   output_chars = <465536>
     iprd[23334]:   session_time = <1a>
     iprd[23334]:   user_ip = <10.0.0.101>
     iprd[23334]:   user_mask = <255.255.255.255>
     iprd[23334]:   acct_type = <STOP>
     iprd[23334]:   net_open: calling
net_connect(auth1.ipass.com, 9101, 1)
     iprd[23334]:  ipass_verify: appl cert,
     isp_code=<Engineering>, ip_addr=<(null)>
     iprd[23334]:   ipass_verify: root cert OK
     iprd[23334]:   net_connect: SSL setup took 2183624
     microseconds
     iprd[23334]:   net_open: Connected to iPass server
     auth1.ipass.com on port 9101
     iprd[23334]:   post_acct_request: done sending
     transaction...  getting reply...
     iprd[23334]:   post_get_status:
     iprd[23334]:   NAME = acct_reply
     iprd[23334]:   tr_seqno = <db1:21328>
     iprd[23334]:   tr_gmt_time = <19-Feb-1997 16:57:03>
     iprd[23334]:   billing_code = <B:9042>
     iprd[23334]:   vnas_list = <192.139.22.2>
     iprd[23334]:   status = <accept>
     iprd[23334]:   post_acct_request: status = accept.
     ./check-ipgen: ipass_remote_acct(fred@ipass.test) = 0,
''
```

Note that full debug output of an authentication daemon, such as``radiusd", can be sent to a file by setting the environment variable ``IPASS_TRACE" to a fully qualified pathname (it must begin with a slash), or to the terminal using the string ``stderr". To include a dump of the results of parsing the configuration file, set the debug_level variable in ipass.conf to ``5".

If you have any problems while installing or testing, contact your assigned support engineer for assistance, or send an e-mail to <support@ipass.com>.

iPass software configuration

iPass configuration file

Name

ipas.conf - iPass configuration file

Description

The iPass configuration file `ipas.conf` contains site specific configuration information used by the iPass Virtual NAS daemon (VNAS) and by the various iPass authentication daemons. A sample configuration file is included in the documentation directory of the iPass distribution.

The location and name of the configuration file may be modified by the use of the config- file option when starting the vnas daemon or the various authentication daemons.

The syntax for the iPass conf file entries is:

```
variable name = value
```

Lines starting with a `#' are ignored and may be used as comments.

The following variable names are currently in use:

Table B-2. The variables and their description are listed below

Variable	**Description**
ssl_profile	The value for this is always isp.pem.
isp_code	This is a unique code that is assigned to each ISP by iPass.
source_ip	This is the IP Address of the machine running vnas.

Table B-2. The variables and their description are listed below

Variable	Description
auth_servers	This is the number of authentication servers available for vnas.
acct_servers	This is the number of accounting servers available for vnas.
auth_server1	This is the IP address of the first RADIUS or TACACS+ authentication server. In most cases this will be the same as source ip.
auth_server2	This is the IP address of the second or alternate authentication server.
acct_server1	The IP address of the first RADIUS or TACACS+ accounting server. In most cases this will the same as source ip.
acc_server2	This is the IP address of the second or alternate accounting server
auth_ port1	The port number that the primary authentication server listens on. Typically 1812 for RADIUS servers and 49 for TACACS+ servers.
auth_port2	The port number that the alternate authentication server listens on.
acct_port1	The port number that the primary accounting server listens on. Typically 1813 for RADIUS servers and 49 for TACACS+ servers.
acct_port_2	The port number that the alternate authentication server listens on
radius_clients	The location of the radius clients file typically, `/etc/raddb/clients.`
tacacs_config	The location and name of the TACACS+ config file.

Table B-2. The variables and their description are listed below

Variable	**Description**
local_domain	The domain to be used for iPass authentication, not necessarily the domain name of this machine.
ipass_servers	The number of iPass servers configured.
ipass_server1	The hostname of the first or primary iPass server typically, `auth1.ipass.com`.
ipass_server2	The hostname of the second or alternate iPass server typically, `auth2.ipass.com.`
ipass_port1	The port number to contact the primary iPass server on, typically 9101
ipass_port2	The port number to contact the alternate iPass server on, typically 9101
default_service	The default service type that the unix-style authentication mode of vnas will return. Valid types are PPP, MPPP, SLIP, CSLIP, LAT, IPX, RLOGIN, TELNET, NASCLI, CLI, FTP, and DIALOUT.
vnas_protocol	The protocol that vnas will use to authenticate users. Valid types for this field are radius, tacacs, unix, and site.
vnas_accounting	Will enable or disable vnas sending accounting to the native accounting daemon. Valid entries for this field are: enable or disable.
local_accounting	Will enable or disable vnas from creating it's own accounting records. Valid entries for this field are enable or disable.
vnas_acct file	This is the pathname and file name that vnas will use to store it's local accounting records

Table B-2. The variables and their description are listed below

Variable	Description
include_domain	Will enable or disable vnas, including domain_name with user-id when vnas passes it to the authentication server. Valid entries for this field are yes or no. The default value is no.
iputmp_enable	This optional flag controls whether or not the vnas daemon will maintain an iputmp field for network user authorization data. Valid entries for this field are: yes or no. The default value is no.
session_limit	The default number of seconds the remote session should be limited to. Note that this value might not be enforced or enforceable by the remote terminal server
time_to_live	The default number of seconds the authentication daemon is to cache the results of successful authentication
auth_cache_size	The number of successful authentication results cached by the authentication server. Each cache entry uses approximately 30 bytes, the is to cache 512 entries. Setting this parameter to 0 turns authentication caching off.
debug_level	This parameter controls the amount of output that is produced when debug tracing is enabled. The default value is ``1". Setting debug_level to ``5" produces the maximum amount of trace output, including a dump of this file.

IPWHO(1)

Name

`ipwho` - display currently active iPass roaming users

Synopsis

ipwho [-aH] [-f file] [user | IP number . . .]

Description

`ipwho` lists the sessions of specified users, and IP numbers. Each line of output contains the user name, the NAS IP address and port number from which the session was initiated, any client IP address, and the start time for the session. If the `-a` flag is specified and the record represents a recently terminated session, the duration of the session is also be shown.

-f file Last - reads the file *file* instead of the default:

`usr/ipass/run/iputmp`

H - displays a header to label the fields.

If multiple arguments are given, the information which applies to any of the arguments is printed. For example, ``ipwho george 10.2.2.1" would list all of ``george's" sessions, as well as all sessions with a client IP address of 10.2.2.1. If no users, or IP numbers are specified, ipwho prints a record of all currently active sessions.

Files

`/usr/ipass/run/iputmp` user authorization data base

See Also: vnas, iputmp, ipgen

Standards

No specific standards are adhered to.

Bugs

The `-a` option is nearly useless on the current accounting record, since by default the first session-end record found in the file is over-written by the next session-start record.

If optional username or IP address arguments select the same record multiple times, each selection is printed. Filter the output through ``sort -u" to eliminate duplicates.

Testing the iPass software

Introduction

This document provides information about how to test the iPass software. It covers initial installation testing, testing with the ipass test center, and testing with other ISPs. Its intended audience is the technical and operations staff of iPass Alliance members.

Note: If you have any problem with installation or testing, contact your assigned support engineer for assistance, or send an email to support@ipass.com.

Test procedures

There are three steps involved in testing the iPass software:

1 **Local testing**

This testing is performed after you have successfully installed the iPass software on your local system. It involves generating test transactions which are handled locally by the iPass software on your system. There is no interaction with the iPass servers during this testing step.

2 **Testing with the iPass server**

In this step, test transactions are generated from your system and sent to the central iPass server. You can also generate transactions from iPass to your system, using an iPass supplied phone number, to verify that transactions are successfully processed by your local system.

3 **Testing with Other Members**

In this step, you can interchange test transactions with other member ISPs. They, in turn, can perform testing with your system.

Testing Local vnas daemon

The first stage in the test process is to verify that the vnas daemon is installed and configured correctly. Included in the /usr/ipass/test directory is a utility called 'check-vnas' that may be used for this purpose.

The syntax for this utility is 'check-vnas username'. Use a local account id for the user name and supply the password for the local account when prompted to do so.

The check-vnas utility contacts the local vnas daemon in the same manner that the central iPass server would. The vnas daemon then contacts the local authentication server to authenticate the user. Diagnostic output showing the transactions and any error messages is displayed on stdout.

After the vnas daemon has been tested and any problems fixed, the next step is to send a transaction to the central iPass server.

Test utility for iPass vnas daemon

Name

check-vnas - Test utility for iPass vnas daemon

Synopsis

check-vnas user id [@ domain]

Description

The `check-vnas` utility is used to verify that the vnas daemon portion of the iPass software is installed correctly.

The utility will connect to the vnas daemon in the same manner as the central iPass server would do. Vnas authenticates the userid using whatever authentication method is in use at the customer site.

Optionally, a domain name may also be provided. If the domain is remote, the authentication daemon will forward the request to the iPass server for authentication. The domain ipas.test may be used to perform a loop- back test. The iPass server will forward the request back to the vnas daemon with the domain portion stripped out. vnas will then perform the appropriate authentication daemon to validate the userid.

See Also: See also vnas, ipass.conf

Standards

No specific standards are adhered to.

Testing with the iPass server

Loopback test

Included in the /usr/ipass/test directory is a utility called 'check-ipgen' that may be used to send transactions for a complete session to iPass. The check-ipgen utility eliminates the need for a NAS for the initial testing which confirms the operational status of the iPass software, and also provides voluminous debug output for trouble-shooting. You will probably want to run it in such a way that you can review many screens full of information after it finishes -- perhaps in an xterm with a fairly large scroll-back buffer.

The syntax for this utility is 'check-ipgen username@ipass.authdom'. For the loopback test, use a local account id for the user name, use ``ipass.test" for the iPass authentication domain name. Supply the password for the local account when prompted to do so. The sequence of events for a loopback test is described below. Copious trace output is produced for each step when you run the test.

- The check-ipgen process acts as a local iPass daemon and forwards the request to the central iPass server. Before doing so, it replaces the authentication domain name in the request with the value configured as the local authentication domain.
- The central iPass server forwards the request back to your local vnas daemon.
- The local vnas daemon authenticates 'username' by contacting the local authentication daemon. Note that the authentication domain has been stripped off to ensure that the local authentication daemon does not recognize this request as an iPass roaming user and thus loop again through the iPass central server.
- The local vnas receives the authorization status from your local authentication daemon and returns it to the central iPass server.
- The central iPass server returns the authorization status to the check-ipgen program.
- The check-ipgen program verifies that a successful authentication was cached locally by re-authenticating the user. The length of time to return the second authorization should negligible.
- The check-ipgen program then sends an accounting session start transaction to the central iPass server.
- The central iPass server logs the session start in the iPass database.
- The central iPass server forwards the accounting session start transaction to the local vnas daemon.
- The local vnas daemon forwards the accounting session start data to the local authentication server for local logging.
- The local authentication server logs the session start data and returns an acknowledgment of this logging to the local vnas daemon.
- The local vnas daemon returns the accounting acknowledgment to the central iPass server.
- The central iPass server returns the accounting acknowledgment to the check-ipgen program.
- Finally, the check-ipgen program sends a simulated accounting session end transaction in exactly the same way as the session start transaction.

Note: A full debug output, including a dump of the results of parsing the configuration file, can be enabled by setting the environment variable ``IPASS_TRACE" to a fully qualified pathname (for example, it must begin with a '/'), or to the string ``stderr".

Dial-up test

iPass maintains a dedicated modem and phone line so that you can perform dial-up testing. The modem is located at iPass's engineering centre in Toronto, Canada, and is connected to a system that simulates an ISP.

The telephone number of this modem is:

905 542 7740

You need to dial the appropriate international codes if you are calling from outside North America.

The modem is configured to accept connections at up to 28.8K.

Your login string in your client software should be configured with a valid user name on your local system, followed by your authentication domain. For example:

```
login:     fred@someplace.net
Password:  xxxxxx
```

When you successfully connect to the modem, iPass will forward authentication transactions to your system.

You should test the following combinations:

1 valid user name and valid password

2 valid user name and incorrect password

3 invalid user name

Verify that you get connected only in the first case (valid user name and valid password.)

Virtual network access server

Name

vnas - virtual network access server

Synopsis

vnas [- c config-file] [-d l] [-t trace file]

Description

`Vnas` implements a virtual network access server using the iPass Alliance protocols. On start-up, vnas determines what protocol to use by reading the ipass.conf file. Valid protocols are RADIUS, TACACS, UNIX Password, and site specific. Normally, vnas is started from inetd.

The following options are available:

-c config-file - specify an alternate configuration file. The standard configuration file is:

```
 /usr/ipass/ipass.conf
```

-d - Turn debug tracing on. Vnas writes trace output to the trace file**.**

-l - long running task. Use this flag when running vnas stand-alone, i.e., not from inetd

-t trace- file - Specify the file to receive trace output. The default is:

```
    /usr/ipass/logs/vnas.trace
```

which is used if the argument supplied is not a file name (for example, ``yes”).

The location of the configuration file can also be specified using the environment variable IPASS_CONFIG. If it is set, it should contain the full path name for the configuration file.

The debug trace can also be turned on using the environment variable IPASS_TRACE. If it is set to ``yes", trace output will go to the default file,

```
/usr/ipass/logs/vnas/trace
```

If it is set to a string starting with a ``/" (slash), it is interpreted as a path name for the file to receive the trace output.

Warning: If it is set to the string ``stderr", trace output is sent to the standard error file. However, since daemon is normally run from inetd, the output is lost down the socket connection (it also corrupts the SSL protocol running on that connection).

The following example should be added to your /etc/inetd.conf file:

```
vnas  stream  tcp  nowait  root  /usr/ipass/bin/vnas vnas
```

Environment

IPASS_CONFIG
optional location of the general configuration file.

IPASS_TRACE
optional location of the trace log file.

IPASS_HOME
installation directory for locating SSL certificate and key files.

Files

`usr/ipass/ipass.conf`
config general configuration file.

`usr/ipass/logs/vnas.trace`
debug and execution trace output.

Standards

No specific standards are adhered to.

iPass accounting

Introduction

This document describes the accounting record format generated by the vnas daemon when it is configured to provide its own accounting records.

The iPass Virtual NAS daemon, or vnas, may be configured to either create its own accounting records, or to pass accounting information to the native accounting server running on the machine, such as RADIUS or TACACS.   It may also be configured to provide both.

Most ISPs prefer to configure vnas to provide its own accounting records so that roaming users are not logged in the same place as the ISPs regular dial-up users. The type of accounting vnas provides is specified in the iPass.conf file that the vnas daemon reads on start-up. The location of the accounting file is also specified in the iPass.conf file.

iPass accounting record format

The following is an example of an iPass accounting record as created by the vnas daemon.

```
Trans_No.      = db1:20981
GMT_Time       = 18-Feb-1997 21:40:21
Billing_code   = B:9042
Date           = Tue Feb 18 16:53:33 EST 1997
Remote_domain  = ipass.com
Remote_ispcode = 102
Session_id     = 1C000013
User_id        = ttest
Nas_ip         = 192.139.22.3
Acct_type      = Stop
Service_type   = PPP
User_ip        = 10.0.0.101
```

```
    User_mask      = 255.255.255.255
    Host_ip        = 192.168.1.2
    Host_port      = 23
    Session_time   = 7
```

The format of the accounting records is white space, followed by a keyword, followed by white space, followed by "=", followed by data. Each line is terminated with a new line character.The various keywords are defined as follows:

Table B-3. The iPass accounting record keywords are described below

Keyword	**Description**
Trans_No	This is a unique transaction number that is supplied by the iPass database server. It is used to enable ISPs to match these accounting records with the monthly accounting statements issued to each member ISP by iPass. The Trans_No. field is the first field in the accounting record.
GMT_Time	The time of the transaction in GMT.
Date	The keyword "Date" followed by the date in the format shown in the above example.
Remote_domain	The remote domain that the user logged into.
Remote_ispcode	The ISP code of the remote domain.
Session_id	The identifier of the session. This field is used to tie START and STOP type accounting records together.
User_id	The userid of the user who used the service.
Nas_ip to.	This is the IP address of the actual NAS that the user connected to.
Host_ip	This is the login host IP address for a telnet/rlogin session.
Host_port	This is the port number on the login host for a telnet/ rlogin session.

Table B-3. The iPass accounting record keywords are described below

Keyword	Description
User_ip	This is the framed user's session IP address.
User_mask	This is the framed user's session netmask.
Acct_type	The type of accounting record. Valid data in the field is either "Start" or "Stop".
Service_type	The type of service that was used. Currently defined types are "Default Service", "PPP", "MPPP", "SLIP", "CSLIP", "LAT", "IPX", "RLOGIN", "TELNET", "NASCLI", "CLI", "FTP", "DIALOUT", and "Unknown Service"
Session_time	This field is always present, however it is only valid for STOP type accounting records. It represents the number of seconds that the user was connected.

iPass call detail record format

Introduction

This document describes the format of the iPass Call Detail Record accounting file. This file is delivered periodically to iPass member ISPs.

The CDR file contains information describing connection time used at remote ISPs by customers of the member ISP. The purpose of the CDR file is to allow member ISPs to generate appropriate end-user information.

File contents

The CDR file is an ASCII file containing one record per connection. Each record is stored in *command delimited* format.

Character strings are surrounded by double quotation marks. Numeric fields may include a decimal point. The fields are as follows:

Table B-4. The iPass call detail record keywords are described below

Keyword	Description
Transaction ID	This field contains a unique identifier for each transaction. It is a character field of up to 32 ASCII characters. It may be used for matching transaction records with other reports, and for identifying transactions when investigating problems.
Billing Code	This field is used internally by iPass. It is a character field of up to 16 ASCII characters.
Billing Rate	A numeric field containing the billing rate in $US per hour for the location at which service was provided.
User ID	This is the user's login ID at the member ISP. The length of this field can be up to 64 ASCII characters.
Authentication Domain	This is the authentication domain specified by the user when connecting. It is normally identical to the home ISP authentication domain, but may be different if an alias domain name is used (for example, mail.isp.net instead of isp.net). The length of this field can be up to 128 ASCII characters.

Table B-4. The iPass call detail record keywords are described below

Keyword	**Description**
Description	This field contains a description of the location at which service was supplied. This is a character field of up to 64 ASCII characters. The contents of this field depend on information supplied by the remote ISP. Possible contents include the name of the remote ISP, the city where service was provided, or the name of the server system which provided service. This field is provided so that ISPs can include a location description in their customer statements
GMT Time	This time stamp is in GMT time and records when the transaction was processed by the iPass server system. It is the time used by iPass for internal processing purposes such as determining which month a transaction occurred in. This is a fixed length character string with format "DD-MON-YYYY HH:MM:SS". Note that 4 digit year codes are used.
Local Time	This time stamp is in local time and records when the transaction was processed by the originating ISP. It records the time when the user's session finished. This field is provided so that member ISPs may include it on end-user statements, as it is more likely to be meaningful to the end-user than the GMT time. Note that the accuracy of this time stamp depends entirely on the accuracy of the clock at the originating ISP. It is a fixed length character string with format "DD-MON-YYYY HH:MM:SS".

Table B-4. The iPass call detail record keywords are described below

Keyword	Description
Length of session	A numeric field indicating the number of seconds that the user was connected.
Net Billing Amount	This is the amount, in $US, that is billed by iPass to the member ISP.

Note: All amounts are in US funds. It is the responsibility of the member ISP billing system to convert the amounts to the appropriate local currency.

Sample data

The following set of records may be used as sample data to test with member ISP billing systems. (An ASCII version of this data may be found in the file *sample cdr.dat .)*

```
"cen:23482","0","fred","anynet.com","Monthly Usage”,
Charge","01-Dec-1996 00:00:00","01-Dec-1996 00:00:00",0,2,2

"db2:2471","0","fred","anynet.com","Korea","02-Dec-1996
02:50:48","02-Dec-1996 02:50:48",444,2.5,.31

"db2:2481","0","fred","anynet.com","Korea","02-Dec-1996
07:41:52","02-Dec-1996 07:41:52",444,2.5,.31

"db2:2559","0","fred","anynet.com","Korea","03-Dec-1996
03:03:38","03-Dec-1996 03:03:38",444,2.5,.31

"db1:6050","0","fred","anynet.com","Korea","04-Dec-1996
17:31:06","04-Dec-1996 17:31:06",15,2.5,.01

"db2:3943","0","fred","anynet.com","Hong Kong","20-Dec-1996
03:50:23","20-Dec-1996 03:48:51",213,3.5,.21

"cen:23483","0","bill","anynet.com","Monthly Usage
Charge","01-Dec-1996 00:00:00","01-Dec-1996 00:00:00",0,2,2

"db1:22540","0","bill","anynet.com","Buffalo, NY","05-Dec-
1996 22:44:22","05-Dec-1996 22:44:22",60,1.6,.03

"db1:22542","0","bill","anynet.com","Buffalo, NY","05-Dec-
1996 23:15:52","05-Dec-1996 23:15:52",60,1.6,.03
```

```
"db1:22450","0","bill","anynet.com","Buffalo, NY","10-Dec-
1996 23:04:24","10-Dec-1996 23:04:24",60,1.6,.03

"db3:22640","0","bill","anynet.com","Buffalo, NY","17-Dec-
1996 15:35:56","17-Dec-1996 15:35:56",60,1.6,.03

"db3:22804","0","bill","anynet.com","Buffalo, NY","24-Dec-
1996 16:12:05","24-Dec-1996 16:12:05",60,1.6,.03

"db3:22885","0","bill","anynet.com","Buffalo, NY","31-Dec-
1996 16:23:01","31-Dec-1996 16:23:01",60,1.6,.03
```

Attributes Reference

C

This chapter contains an alphabetic list of NavisRadius attributes. The list's entries describe the attributes and the values you may assign to them. Each entry includes a section which describes the attribute and explains its use. The list contains all the attributes defined in the original RADIUS protocol, although the majority of the attributes are Ascend's vendor-specific additions to RADIUS.

Note: Because many of the attributes are Ascend vendor-specific, they are based on parameters for Ascend NAS products. The attribute descriptions and usage explanations of Ascend-specific entries refer to the MAX or the MAX TNT. You may obtain more information about these parameters by consulting the manuals and supplements that ship with your specific Ascend NAS gear.

Attribute Name

Description: The Description text explains the attribute.

Usage: The Usage text explains the values you can specify for the attribute.

Example: The Example text presents an example of how to use the attribute.

Dependencies: The Dependencies text tells you what other information you need in order to specify the proper value for the attribute.

Acct-Authentic (45)

Description: This attribute specifies whether an incoming call was authenticated by RADIUS, TACACS, or a local Connection Profile, or whether the MAX accepted the call without authentication.

Acct-Authentic is sent in an Accounting-Request packet under these conditions:

- At the start of a session (when Acct-Status-Type=Start)
- At the end of an authenticated session (Acct-Status-Type=Stop) when the Auth parameter is not set to RADIUS/LOGOUT

Usage: Acct-Authentic does not appear in a user profile. It can have either of the following values:

- RADIUS (1)

  This value indicates that RADIUS authenticated the incoming call. RADIUS is the default.
- Local (2)

  This value indicates that an incoming call was authenticated by a local Connection Profile or by TACACS, or that the call was accepted without authentication.

Acct-Delay-Time (41)

Description: This attribute specifies how many seconds the MAX has been trying to send this Accounting packet.

Acct-Delay-Time is sent in an Accounting-Request packet under these conditions:

- At the start of a session (when Acct-Status-Type=Start)
- At the end of a session or when a session fails to authenticate (Acct-Status-Type=Stop) and the Auth parameter is not set to RADIUS/LOGOUT

Usage: Acct-Delay-Time does not appear in a user profile. Its default value is 0 (zero).

Acct-Input-Octets (42)

Description: This attribute specifies how many octets have been received during the session.
Description: Acct-Input-Octets is sent in an Accounting-Request packet at the end of a session (Acct-Status-Type=Stop) when both of these conditions are true:

- The session has been authenticated.
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Acct-Input-Octets does not appear in a user profile. Its default value is 0 (zero).

Acct-Input-packets (47)

Description: This attribute specifies how many packets have been received during the session. Acct-Input-packets is sent in an Accounting-Request packet at the end of a session (Acct-Status-Type=Stop) when all of these conditions are true:

- The session has been authenticated.
- The Auth parameter is not set to RADIUS/LOGOUT.
- A framed protocol is in use.

Usage: Acct-Input-packets does not appear in a user profile. Its default value is (zero).

Acct-Output-Octets (43)

Description: This attribute specifies how many octets have been sent during the session.

Acct-Output-Octets is sent in an Accounting-Request packet at the end of a session
(Acct-Status-Type=Stop) when both of these conditions are true:

- The session has been authenticated.
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Acct-Output-Octets does not appear in a user profile. Its default value is (zero).

Acct-Output-packets (48)

Description: This attribute specifies how many packets have been sent during the session. Acct-Output-packets is sent in an Accounting-Request packet at the end of a session
(Acct-Status-Type=Stop) when all of these conditions are true:

- The Auth parameter is not set to RADIUS/LOGOUT.
- The session is authenticated.
- A framed protocol is in use.

Usage: Acct-Output-packets does not appear in a user profile. Its default value is (zero).

Acct-Session-Id (44)

Description: This attribute specifies a unique numeric string identified with the bridging, routing, or terminal server session reported in the Accounting-Request packet. RADIUS correlates the Accounting Start packet and Accounting Stop packet using Acct-Session-Id.

Acct-Session-Id is sent under these conditions:

- At the start of a session (when Acct-Status-Type=Start)
- At the end of a session or when a session failed to authenticate (Acct-Status-Type=Stop) and the Auth parameter is not set to RADIUS/LOGOUT

Usage: Acct-Session-Id does not appear in a user profile. Its value can range from 1 to 2,137,383,647. For every session, RADIUS generates a unique session ID, thereby preventing the same session ID from being used for more than one session.

Dependencies: Keep this additional information in mind:

- SNMP accounting uses session reference numbers to identify sessions; when an SNMP accounting session and a RADIUS accounting session have the same ID, they are identical.

However, SNMP records all calls, while RADIUS records only those calls that result in a successful login or authentication.

- Using the Acct-ID Base parameter in the Ethernet Profile, you can specify whether the numeric base of the Acct-Session-Id attribute is 10 or 16.

  This parameter controls how the Acct-Session-Id attribute is presented to the accounting server. For more information, see the *MAX Reference Guide*.
- The Acct-Session-Id attribute is defined in section 5.5 of IETF RFC 2059 for RADIUS accounting.

Acct-Session-Time (46)

Description: This attribute specifies how many seconds the session has been logged in.

Acct-Session-Time is sent in an Accounting-Request packet at the end of a session (Acct-Status-Type=Stop) when both of these conditions are true:

- The session has been authenticated.
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Acct-Session-Time does not appear in a user profile. Its default value is 0 (zero).

Acct-Status-Type (40)

Description: This attribute specifies whether the Accounting packet sent to the RADIUS server is the beginning (Start) or end (Stop) of a bridging, routing, or terminal server session.

Acct-Status-Type is included under these conditions:

- At the start of a session (when Acct-Status-Type=Start)
- At the end of a session or when a session fails to authenticate (when Acct-Status-Type=Stop), and only if the Auth parameter is not set to RADIUS/ LOGOUT

Usage: Acct-Status-Type does not appear in a user profile.

Ascend-Add-Seconds (240)

Description: This attribute specifies the number of seconds that average line utilization (ALU) for transmitted data must exceed the threshold indicated by the Ascend-Target-Util attribute before the MAX begins adding bandwidth to a session. The MAX determines the ALU for a session by using the algorithm specified by the Ascend-History-Weigh-Type attribute.

When utilization exceeds the threshold for a period of time greater than the value of the Ascend-Add-Seconds attribute, the MAX attempts to add the number of channels specified by the Ascend-Inc-Channel-Count attribute. Using the Ascend-Add-Seconds and Ascend-Remove-Seconds attributes prevents the system from continually adding and subtracting bandwidth, and can slow down the process of allocating or removing bandwidth.

Usage: Specify a number between 1 and 300. The default value is 5.

Dependencies: Keep this additional information in mind:

- Additional channels must be available, and the number of channels added cannot exceed the amount specified by the Ascend-Maximum-Channels attribute.
- Ascend-Add-Seconds and Ascend-Remove-Seconds have little or no effect on a system with a high Ascend-Seconds-Of-History value.

  If the value of Ascend-Seconds-Of-History is low, the Ascend-Add-Seconds and Ascend-Remove-Seconds attributes provide an alternative way to ensure that spikes must persist for a certain period of time before the system responds.

Ascend-Appletalk-Peer-Mode (117)

Description: Specifies whether the connection is for a single dial-in station or for a router.

Usage: Specify one of the following values:

- Appletalk-Peer-Router (0) specifies that the caller is an AppleTalk router, such as an Ascend Pipeline unit.
- Appletalk-Peer-Dialin specifies that the caller is a dial-in AppleTalk client, such as a single Macintosh dialing in over a modem.

Example: The following example shows a RADIUS user profile for a routed connection:

```
pipe50 Password="pipe50"

   Service-Type= Framed,

   Framed-Protocol = PPP,

   Ascend-Appletalk-Peer-Mode = Appletalk-Peer-Router,

   Ascend-Route-Appletalk = Route-Appletalk-Yes,

   Ascend-Idle-Limit = 0
```

The following is an example of a RADIUS user profile for a dial-in connection:

```
mac1 Password = "mac1"
   Service-Type= Framed,
   Framed-Protocol = PPP,
   Ascend-Appletalk-Peer-Mode = Appletalk-Peer-Dialin,
   Ascend-Route-Appletalk = Route-Appletalk-Yes,
   Ascend-Idle-Limit = 0
```

Dependencies: Ascend-Route-Appletalk must be set to Ascend-Route-Appletalk-Yes.

See Also: "Ascend-Appletalk-Peer-Mode (117)." "Ascend-Appletalk-Route (116)."

Ascend-Appletalk-Route (116)

Description: Defines a static AppleTalk route in a RADIUS pseudo-user profile.

Usage: Create a pseudo-user profile with the first line in the following format:

```
appleroute- num Password="ascend', user-service=Outbound
```

where *num* is a number in a series starting at 1. Then enter one or more static AppleTalk route specifications in the following format:

```
Ascend-Appletalk-Route=" net_start net_end zone_name profile
name
```

Table C-1. This table lists the route arguments and their description

Argument	Description
<net_start>	The lower limit of the network range for this network. A network range is a range of network numbers set into the port descriptor of the router port and then transmitted through RTMP to the other nodes of the network. Each of the numbers within a network range can represent up to 253 devices. The default is blank
<net_end>	The upper limit of the network range for this network. This range defines the networks available for packets routed using the static route. Specify a number between 1 and 65199. If there are other AppleTalk routers on the network, you must configure the network ranges to be identical to the ranges specified on the other routers.
<zone_name>	The name of the AppleTalk zone associated with this network. A zone is a multicast address containing a subset of the AppleTalk nodes on 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 case-insensitive. However, because some routers regard zone names as case-sensitive, the spelling of zone names should be consistent when you configure multiple connections or routers You can use up to 33 alphanumeric characters. The default is blank.

Table C-1. This table lists the route arguments and their description

Argument	Description
<profile_name>	The outgoing RADIUS user profile that the route uses. The default is blank.

Each static route must appear in a user profile. User profile entries for Appletalk static routes are identified by the special name appleroute-# and have the following format:

```
appleroute-# Password = “ascend” Service-Type= Outbound

Address 1

Address 2

...

Address n
```

Address *n* is the actual route associated with this entry.

An example of a static route with the associated connection profiles is:

```
appleroute-1 Password = “ascend” Service-Type= Outbound-
User Ascend-Appletalk-Route = “20 25 testzone1 pipe50”


pipe50 Password = “ascend” Service-Type= Outbound,
   Service-Type= Framed,
   Framed-Protocol = MPP,
   Ascend-Appletalk-Peer-Mode = Appletalk-Peer-Router,
   Ascend-Route-Appletalk = Route-Appletalk-Yes,
   Ascend-Dialout-Allowed = Dialout-Allowed,
   Ascend-Dial-Number = “83272”,
   Ascend-Send-Auth = Send-Auth-PAP,
   Ascend-Send-Passwd = “MAX”
```

Dependencies: Ascend-Route-Appletalk must be set to Yes.

See Also: Ascend-Appletalk-Peer-Mode (117)

Ascend-Ara-PW (181)

Note: This attribute no longer appears in a user profile. The user profile at the end of this description does illustrate how you configure a user profile to specify the password of an incoming caller over AppleTalk Remote Access (ARA).

Description: This attribute specifies the password of the incoming caller over ARA (AppleTalk Remote Access). The ARA software in the MAX uses DES to encrypt and decrypt the password.

Example: This example sets up a TCP connection through ARA with dynamic IP address assignment:

```
Emma Authentication-Type=ARA-DES, Password=”pwd”
     Framed-Protocol=Ascend-ARA,
     Ascend-Send-Secret="pwd",
     Ascend-Route-IP=Route-IP-Yes,
     Ascend-Assign-IP-Pool=1
```

Ascend-Assign-IP-Client (144)

Description: In the Radipa-Hosts pseudo-user profile, the Ascend-Assign-IP-Client attribute specifies the IP address of an Ascend unit that can use global IP address pools.

Usage: Specify an IP address in dotted-decimal notation. The default value is 0.0.0.0. You can specify multiple instances of this attribute. At present, the MAX does not use the list of radipad client units.

Dependencies: If no Ascend-Assign-IP-Client attribute is present, the list of client units defaults to those present in the RADIUS clients file.

See Also: “Ascend-Assign-IP-Global-Pool (146).”
“Ascend-Assign-IP-Server (145.”

Ascend-Assign-IP-Global-Pool (146)

Description: In a RADIUS user profile requiring dynamic addressing for dial-in users, the Ascend-Assign-IP-Global-Pool attribute specifies the global address pool from which RADIUS should assign each user an address.

Usage: Specify the name of the pseudo-user profile containing global IP pool definitions. The Ascend unit tries to allocate an address from the pools in order, and chooses an address from the pool with the first available IP address.

Dependencies: Do not set the Framed-IP-Address attribute in the user profile. If you do, the MAX will require the caller to use the static IP address the attribute specifies.)

See Also: “Ascend-Assign-IP-Client (144).” “Ascend-Assign-IP-Server (145).” “Framed-IP-Address (8).”

Ascend-Assign-IP-Pool (218)

Description: This attribute specifies the address pool from which RADIUS assigns the user an IP address.

A pool is a range of contiguous IP addresses on your local network. The MAX chooses an address from these pools and assigns it to an incoming call when Assign Adrs=Yes in the Answer Profile, or when the calling station requests an address assignment. Assigning an address to a device is called performing dynamic IP. Dynamic IP can apply when the calling end is a station; however, if the calling end is a router, that router usually rejects attempts to perform dynamic IP.

If you need to define more than two pools of addresses, you must use the RADIUS attribute Ascend-IP-Pool-Definition to configure the IP address pools.

Usage: Specify an integer corresponding to an address pool. The default value is 1.

Example: In the user profile, the host is configured to request an address from address pool #2:

```
emma Password="m2dan"
        Service-type=Framed
```

```
Framed-Protocol=PPP,
Ascend-Route-IP=Route-IP-Yes,
Ascend-Metric=2,
Framed-Routing=None,
Ascend-Assign-IP-Pool=2
```

Ascend-Assign-IP-Server (145)

Description: In the Radipa-Hosts pseudo-user profile, the Ascend-Assign-IP-Server attribute specifies the IP address of the host running radipad.

Usage: Specify an IP address in dotted decimal notation. The default value is 0.0.0.0. Only one instance of this attribute can appear in the profile. The default value is a placeholder only. You must specify a valid IP address for radipad to work.

See Also: "Ascend-Assign-IP-Client (144)." "Ascend-Assign-IP-Global-Pool (146)."

Ascend-ATM-Vci (95)

Description: Description: Specifies the Virtual Channel Identifier for an ATM connection.

Usage: Specify a value from 32 to 1023. The default is 32. The maximum setting is determined by MAX TNT hardware capabilities.

Example: The following sample profile specifies Frame Relay to ATM switching:

```
permconn-yossi-1 Password="ascend"
   User-Service=Outbound,
   Framed-Protocol=ATM-FR-CIR,
   User-Name="atm-30-sw",
   Ascend-Metric=2,
   Framed-Routing=None,
   Ascend-Idle-Limit=30,
   Ascend-Group="70",
   Acct-Authentic=None,
   Ascend-Send-Auth=Send-Auth-None,
   Ascend-Call-Type=Nailed,
   Ascend-FT1-Caller=FT1-Yes,
   Ascend-Route-IP=Route-IP-No,
   Ascend-ATM-Vpi=1,
   Ascend-ATM-Vci=43,
   Ascend-FR-Circuit-Name="adsl-atm",
   Ascend-Data-Svc=Nailed-64K
```

See Also: “Ascend-ATM-Vpi (94).” “Framed-Protocol (7).”

Ascend-ATM-Vpi (94)

Description: Specifies the Virtual Path Identifier for an ATM connection.

Usage: Specify a value from 0 to 15. The default is 0 (zero).

Example: The following sample profile specifies ATM encapsulation:

```
permconn-yossi-2 Password="ascend"

User-Service=Outbound,
   Framed-Protocol=ATM-1483,
   Framed-IP-Address=222.222.222.1,
   Framed-IP-Netmask=255.255.255.0,

User-Name="atm-30",

   Ascend-Metric=2,
   Framed-Routing=None,
   Ascend-Idle-Limit=30,
   Ascend-Group="70",
   Acct-Authentic=None,
   Ascend-Send-Auth=Send-Auth-None,
   Ascend-Call-Type=Nailed,
   Ascend-FT1-Caller=FT1-Yes,
   Ascend-Route-IP=Route-IP-Yes,
   Ascend-ATM-Vpi=1,
   Ascend-ATM-Vci=42,
   Ascend-Data-Svc=Nailed-64K
```

See Also: “Ascend-ATM-Vci (95).”
“Framed-Protocol (7).”

Ascend-Authen-Alias (203)

Description: This attribute sets the MAX unit's login name during PPP authentication.

When the MAX places an outgoing call, it identifies itself by a login name and password. The login name is either its system name (as specified by the Name parameter in the System Profile) or the value you specify for the Ascend-Authen-Alias attribute.

Usage: Specify a text string containing up to 16 characters. The default is the value of the Name parameter in the System Profile.

Example: This example uses the Ascend-Authen-Alias attribute in an outgoing profile:

```
homer-out Password="ascend", Service-Type=Outbound
        User-Name="homer",
        Ascend-Authen-Alias="myMAXcallingU",
        Ascend-Send-Auth=Send-Auth-PAP,
        Ascend-Send-Secret="passwrd1",
        Ascend-Dial-Number="31",
        Framed-Protocol=PPP,
        Framed-IP-Address=10.0.100.1,
        Framed-IP-Netmask=255.255.255.0,
        Ascend-Metric=2,
        Framed-Routing=None,
        Framed-Route="10.5.0.0/24 10.0.100.1 1",
        Ascend-Idle-Limit=30
```

Ascend-Backup (176)

Description: This attribute specifies the name of a backup profile for a nailed-up link when the physical connection fails. The MAX automatically diverts traffic to the backup connection. When the primary connection is restored, traffic again uses the primary connection.

When you use the backup connection, the MAX does not move routes to the backup profile. Therefore, the IP routes shown in the terminal server display may be incorrect, although statistical counts reflect the change.

Usage: Specify the name of the profile that you want to act as the backup. The backup connection can be switched or nailed up. The default value is null.

Dependencies: Keep this additional information in mind:

- Do not create nested backup connections.
- Attributes that you define for the primary profile do not automatically apply to the backup profile.

  For example, if you set the primary profile to filter Telnet packets, you must set the backup profile to filter Telnet packets as well. Outgoing Frame Relay packets are the only packets that follow the primary profile definitions. All other packets follow the backup profile definitions.

Ascend-BACP-Enable (133)

Description: The Ascend-BACP-Enable attribute specifies whether Bandwidth Allocation Control Protocol (BACP) is enabled for the link.

BACP is the Internet standard protocol equivalent to the Ascend MP+ bandwidth allocation protocol. BACP functions similarly to MP+ and uses the same attributes as MP+.

Usage: Usage: You can specify one of these settings:

- BACP-No (0) disables BACP for the link.

  The default value is BACP-No.
- BACP-Yes (1) enables BACP for the link.

Ascend-Base-Channel-Count (172)

Description: This attribute specifies the initial number of channels the MAX sets up when originating calls for a PPP, MP+, MP, or Combinet multichannel link.

Usage: The maximum number of channels you can specify depends upon the nature of the link:

- For a PPP link, the maximum number of channels is always 1.

- For an MP+ or MP link, the amount you specify is limited by the number of channels available, but the device at the remote end of the link must also support MP+ or MP.
- For a Combinet link, you can specify up to two channels.

The default value is 1.

Dependencies: Keep this additional information in mind:

- The Ascend-Base-Channel-Count attribute does not apply when all channels of the link are nailed up (Ascend-Call-Type=Nailed).
- For optimum MP+ performance, both sides of a connection must set these parameters and attributes to the same values:
  - Base Ch Count (in the Connection Profile) or Ascend-Base-Channel-Count (in RADIUS)
  - Min Ch Count (in the Answer Profile and the Connection Profile) or Ascend-Minimum-Channels (in RADIUS)
  - Max Ch Count (in the Answer Profile and the Connection Profile) or Ascend-Maximum-Channels (in RADIUS)

Ascend-Billing-Number (249)

Description: This attribute specifies a billing number for charges incurred on the line. If you do not enter a billing number, the telephone company bills charges to the telephone number assigned to the line.

Your carrier determines the billing number, and uses it to sort your bill. If you have several departments, and each department has its own Ascend-Billing-Number, your carrier can separate and tally each department's usage.

Usage: Specify a telephone number. You can indicate up to ten characters, and you must limit those characters to the following:

```
1234567890()[]!z-*# |
```

Dependencies: The MAX uses the Ascend-Billing-Number attribute differently depending on the type of line you use:

- For a T1 line, the MAX appends the value specified in the Ascend-Billing-Number attribute to the end of each phone number it dials for the call.

- Ascend-Billing-Number for outgoing calls on an ISDN BRI line applies only to installations in Australia.
- For a T1 PRI line, the MAX uses the Ascend-Billing-Number rather than the phone number ID to identify itself to the answering party.

  The Clid Auth parameter enables you to require a device to authenticate incoming calls by checking the calling party's phone number. The device performs CLID (Calling Line ID) authentication before answering an incoming call. The calling party's phone number must match the Calling # parameter or the Calling-Station-ID attribute. If the device cannot authenticate the call when CLID authentication is required, the call is rejected.

  If the calling party uses the Ascend-Billing-Number attribute instead of its phone number as its ID, the CLID used by the answering side is not the true phone number of the caller. This situation presents a security breach if you use Clid Auth.

  Further, be aware that if you specify a value for the Ascend-Billing-Number attribute, there is no guarantee that the phone company will send it to the answering device.

Ascend-Bridge (230)

Description: This attribute enables or disables protocol-independent bridging for the user profile.

Usage: You can specify one of these values:

- Bridge-No (0)

  This setting disables bridging for the link. Bridge-No is the default.
- Bridge-Yes (1)

  This setting enables bridging for the link.

Example: This user profile specifies an IPX bridging link:

```
MAX1 Password="m2dan", Service-Type=Framed
     Framed-Protocol=PPP,
     Ascend-Route-IPX=Route-IPX-No,
     Ascend-Bridge=Bridge-Yes,
     Ascend-Handle-IPX=Handle-IPX-Client,
     Ascend-Netware-timeout=30
```

Ascend-Bridge-Address (168)

Description: This attribute specifies the IP address and associated MAC address of a device on a remote LAN to which the MAX can form a bridging connection.

Usage: The Ascend-Bridge-Address attribute has this format:

```
Ascend-Bridge-Address="<MAC_address> <IP_address>"
```

Table C-2 describes Ascend-Bridge-Address arguments.

Table C-2. Ascend-Bridge-Address arguments

Argument	Description
<MAC_address>	Specifies a MAC address in standard 12-digit hexadecimal format (yyyyyyyyyyyy) or in colon-separated format (yy:yy:yy:yy:yy:yy). If the leading digit of a colon-separated pair is 0 (zero), you do not need to enter it; that is, ":y" is the same as ":0y". The default value is 000000000000.
<IP_address>	Specifies an IP address in dotted decimal format. The default value is 0.0.0.0.

When your MAX receives an ARP request for one of the IP devices you specify, the MAX replies with the corresponding MAC address. Because the MAX replies to these ARP requests as if the IP devices were local, you must have user profiles that bridge IP packets to each device.

Dependencies: Each bridge entry must appear in a pseudo-user profile. You create a pseudo-user to store information that the MAX can query—in this case, in order to store bridging information. For a unit-specific bridge entry, specify the first line of a pseudo-user entry in this format:

```
bridge-<unit_name>-<num> Password="ascend", Service-Type=
Outbound
```

<unit_name> is the system name of the MAX—that is, the name specified by the Name parameter in the System Profile. <num> is a number in a sequential series, starting at 1.

In each pseudo-user profile, you specify one or more Ascend-Bridge-Address attributes. When you have properly configured the profile, RADIUS adds bridging entries to the bridge table whenever you power on or reset the MAX, or when you select the Upd Rem Cfg command from the Sys Diag menu. RADIUS adds the entries in this way:

1. RADIUS looks for entries having the format bridge-<unit_name>-<num>, where <unit_name> is the system name and <num> is a number in a sequential series, starting with 1.
2. RADIUS loads the data to create the bridging tables.

Example: This example creates two bridging table entries.

```
bridge-Ascend-1 Password="ascend", Service-Type=Outbound
        Ascend-Bridge-Address="2:2:3:10:11:12 1.2.3.4 1",
        Ascend-Bridge-Address="2:2:3:13:14:15 5.6.7.8 2"
```

Ascend-Call-Attempt-Limit (123)

Description: Specifies how many unsuccessful dial-in attempts can occur before the MAX blocks further connection attempts.

Usage: Specify an integer. The default is 0 (zero), which doubles call blocking.

Example: `Ascend-Call-Attempt-Limit=10`

See Also: "Ascend-Call-Block-Duration (124)"

Ascend-Callback (246)

Description: This attribute enables or disables callback. Callback occurs when the MAX answers a call and verifies a name and password against a user profile. If
Ascend-Callback=Yes, the MAX hangs up and dials back to the caller using these values:

- The phone number specified by Ascend-Dial-Number
- The password specified by Ascend-Send-Secret or Ascend-Send-Passwd
- Any other relevant attributes in the user profile that authenticated the call

Note: If you set up a RADIUS user profile for callback and CLID-only authentication, the MAX never answers the call; the caller can therefore avoid billing charges.

See Also: You can specify one of these values:

- Callback-No (0)

  This value indicates that the MAX answers in the normal manner after authentication.
- Callback-Yes (1)

  This value indicates that the MAX hangs up and calls back the caller after authentication.

Dependencies: The Ascend-Callback attribute applies only to incoming calls and should not appear in dial-out user profiles (when Service-Type=Outbound).

Ascend Callback-Delay (108)

Description: Specifies the number of seconds the MAX TNT waits before calling back a remote user.

Usage: Specify an integer from 0 through 60. The unit treats values of 0-3 as 3 seconds. The default is 0 (zero)

Dependencies: If Ascend-Callback-Callback-No, Ascend-Callback-Delay does not apply.

See Also: "Ascend-Callback (246)."

Ascend-Call-Block-Duration (124)

Description: Specifies the period (in seconds) during which the MAX will refuse connection attempts after the Ascend-Call-Attempt-Limit has been reached. When the timer expires, the MAX will accept further dial-in-attempts, up to the limit specified by Ascend-Call-Attempt-Limit.

Usage: Specify and integer. The default is 0 (zero).

Example: `Ascend-Call-Block-Duration=60`

Dependencies: For Ascend-Call-Call-Block-Duration to apply, you must set Ascend-Call-Attempt-Limit to a non-zero value.

See Also: “Ascend-Call-Attempt-Limit (123)”

Ascend-Calling-Subaddress (107)

Description: Specifies the ISDN subaddress that the MAX send to RADIUS during CLID authentication. The value of Ascend-Calling-Subaddress appears in Access-Request and Accounting Start packets.

Usage: Specify a subaddress. The default is null.

Ascend-Call-By-Call (250)

Description: This attribute specifies the T1 PRI service that the MAX uses when placing a PPP call.

Usage: Specify a number corresponding to the type of service the MAX uses. The default value is 6. Table C-3 lists the services available for each service provider.

Table C-3. Ascend-Call-By-Call settings

Number	AT&T	Sprint	MCI
0	Disable call-by-call service.	Reserved	N/A
1	SDN (including GSDN)	Private	VNET/Vision
2	Megacom 800	Inwatts	800
3	Megacom	Outwatts	PRISM1, PRISM II, WATS
4	N/A	FX	900
5	N/A	Tie Trunk	DAL
6	ACCUNET Switched Digital Services	N/A	N/A

Table C-3. Ascend-Call-By-Call settings

Number	AT&T	Sprint	MCI
7	Long Distance Service (including AT&T World Connect)	N/A	N/A
8	International 800 (I800)	N/A	N/A
16	AT&T MultiQuest	N/A	N/A

Ascend-Call-Filter (243)

Description: Unlike the Filter Profiles in the MAX configuration interface, RADIUS filters are part of the outgoing or incoming RADIUS user profile. In other words, within any RADIUS users file defining a user profile, you can include values for Ascend-Call-Filter to define call filters for that profile. RADIUS filters are retrieved only when a call is placed using a RADIUS outgoing profile or answered using a RADIUS incoming profile.

Usage: Filter entries apply on a first-match basis. Therefore, the order in which filter entries are entered is significant.

If you make changes to a filter in a RADIUS user profile, the changes do not take effect until a call uses that profile.

IP call filter entries

Use this format for an IP call filter entry:

```
Ascend-Call-Filter="ip <dir> <action>
[dstip <dest ipaddr>\<subnet mask>][srcip <src ipaddr>\<sub-
net mask>]
[<proto> [dstport <cmp> <value>] [srcport <cmp> <value>]
[<est>]]"
```

Note: A filter definition cannot contain new lines. The syntax is shown on multiple lines here for printing purposes only.

Table C-4 Describes each element of the syntax. None of the keywords are case sensitive. .

Table C-4. IP call filter syntax elements

Keyword or argument	Description
ip	The keyword “ip” indicates an IP filter.
<dir>	The <dir> argument indicates filter direction. You can specify “in” (to filter packets coming into the MAX) or “out” (to filter packets going out of the MAX).
<action>	<action> indicates what action the MAX should take with a packet that matches the filter. You can specify either “forward” or “drop”.
dstip <dest ipaddr>	“dstip” is a keyword indicating “destination IP address.” The filter applies to packets whose destination address matches the value of <dest ipaddr>. If a subnet mask portion of the address is present, the MAX compares only the masked bits. If you set <dest ipaddr> to 0.0.0.0, or if this keyword and its IP address specification are not present, the filter matches all IP packets.
srcip <src ipaddr>	“srcip” is a keyword indicating “source IP address.” The filter applies to packets whose source address matches the value of <src ipaddr>. If a subnet mask portion of the address is present, the MAX compares only the masked bits. If you set <src ipaddr> to 0.0.0.0, or if this keyword and its IP address specification are not present, the filter matches all IP packets.
<proto>	<proto> indicates a protocol that you can specify as a name or a number. The filter applies to packets whose protocol field matches this value.The supported names and numbers are icmp (1), tcp (6), udp (17), and ospf (89). If you set <proto> to 0 (zero), the filter matches any protocol.

Table C-4. IP call filter syntax elements

Keyword or argument	Description
dstport <cmp> <value>	“dstport” is a keyword indicating “destination port.” This argument is valid only when the protocol is tcp (6) or udp (17). If you do not specify a destination port, the filter matches any port. <cmp> is an argument indicating how to compare the specified value to the actual destination port. It can have the value <, =, >, or !=. <value> can be a number or a name. Supported names and numbers are ftp-data (20), ftp (21), telnet (23), smtp (25), nameserver (42), domain (53), tftp (69), gopher (70), finger (79), www (80), kerberos (88), hostname (101), nntp (119), ntp (123), exec (512), login (513), cmd (514), and talk (517).
srcport <cmp> <value>	“srcport” is a keyword indicating “source port.” It is valid only when the protocol is tcp (6) or udp (17). If you do not specify a source port, the filter matches any port. <cmp> is an argument indicating how to compare the specified value to the actual source port. It can have the value <, =, >, or !=. <value> can be a number or a name. Supported names and numbers are ftp-data (20), ftp (21), telnet (23), smtp (25), nameserver (42), domain (53), tftp (69), gopher (70), finger (79), www (80), kerberos (88), hostname (101), nntp (119), ntp (123), exec (512), login (513), cmd (514), and talk (517).
<est>	If you set this argument to 1, the filter matches a packet only if a TCP session is already established. It is valid only when the <proto> specification is tcp (6).

Generic call filter entries

Use this format for a generic call filter entry:

```
Ascend-Call-Filter="generic <dir> <action> <offset> <mask>
<value> <compare> [<more>]"
```

Note: A filter definition cannot contain new lines. The syntax is shown on multiple lines here for printing purposes only.

Table C-5 describes each element of the syntax. None of the keywords are case sensitive.

Table C-5. Generic call filter syntax elements

Keyword or argument	Description
generic	The keyword "generic" indicates a generic filter.
<dir>	The <dir> argument indicates filter direction. You can specify "in" (to filter packets coming into the MAX) or "out" (to filter packets going out of the MAX).
<action>	<action> indicates what action the MAX should take with a packet that matches the filter. You can specify either "forward" or "drop".
<offset>	<offset> indicates the number of bytes masked from the start of the packet. The byte position specified by <offset> is called the byte-offset. Starting at the position specified by <offset>, the MAX applies the value of the <mask> argument. A mask hides the part of a number that appears behind the binary zeroes in the mask; for example, if you set <mask> to ffff0000 in hexadecimal format, the filter uses only the first 16 binary digits in the comparison, since f=1111 in binary format. The unit then compares the unmasked portion of the packet with the value specified by the <value> argument.

Table C-5. Generic call filter syntax elements

Keyword or argument	Description
<mask>	This argument indicates which bits to compare in a segment of the packet; the mask cannot exceed 6 bytes (12 hexadecimal digits). A one-bit in the mask indicates a bit to compare; a zero-bit indicates a bit to ignore. The length of the mask specifies the length of the comparison.
<value>	This argument indicates the value to compare to the packet contents at the specified offset in the packet. The length of the value must be the same as the length of the mask; otherwise, the MAX ignores the filter.
<compare>	This argument indicates how the MAX compares a packet's contents to the value specified by <value>. You can specify == or !=, for Equal or NotEqual. The default value is Equal.
<more>	If present, this argument specifies whether the MAX applies the next filter definition in the profile to the current packet before making the Forward or Drop decision. The <dir> and <action> values of the next entry must be the same as the <dir> and <action> of the current entry; otherwise, the MAX ignores the <more> flag.

Example: These are examples of IP call filter entries:

```
Ascend-Call-Filter="ip in drop"
```

```
Ascend-Call-Filter="ip out forward tcp"
```

```
Ascend-Call-Filter="ip out forward tcp dstip 10.0.200.3/16
srcip 10.0.200.25/16 dstport!=telnet"
```

```
Ascend-Call-Filter="ip out forward tcp dstip 10.0.200.3/16
srcip 10.0.200.25/16 icmp"
```

Note: A filter definition cannot contain new lines. The syntax is shown on multiple lines here for printing purposes only.

These are examples of generic call filter entries:

```
Ascend-Call-Filter="generic in drop 0 ffff 0080"
Ascend-Call-Filter="generic in drop 0 ffff != 0080 more"
Ascend-Call-Filter="generic in drop 16 ff aa"
```

Ascend-Call-Type (177)

Description: This attribute specifies the type of nailed-up connection in use.

Usage: You can specify one of these values:

- Nailed (1)

  This setting indicates a link that consists entirely of nailed-up channels. Nailed (1) is the default.

- Nailed/Mpp (2)

  This setting indicates a link that consists of both nailed-up and switched channels. The MAX establishes this connection whenever any of its nailed-up or switched channels are connected end-to-end. If a Nailed/Mpp link is down and the nailed-up channels are down, the link cannot re-establish itself until the MAX brings up one or more of the nailed-up channels, or dials one or more switched channels.

  Typically, the switched channels are dialed when the MAX receives a packet whose destination is the unit at the remote end of the Nailed/Mpp connection. The packet initiating the switched call must come from the caller side of the connection.

  If a failed channel is in the group specified by the Ascend-Group attribute, that channel is replaced with a switched channel, even if the call is on-line with more than the minimum number of channels. Failed nailed-up channels are replaced by switched channels, regardless of the Min Ch Count setting.

- Perm/Switched (3)

  This setting indicates a permanent switched connection.

  A permanent switched connection is an outbound call that attempts to remain up at all times. If the unit or central switch resets, or if the link is terminated, the permanent switched connection attempts to restore the link at ten-second intervals.

  Use this setting if your telephone company charges for each incoming and outgoing connection attempt, but does not charge for connection time on local calls. Ascend's regular bandwidth-on-demand feature conserves connection time but causes many connection attempts. A permanent switched connection performs the opposite function—it conserves connection attempts but causes a long connection time.

For the answering device at the remote end of the permanent switched connection, we recommend that the Connection Profile be configured to answer calls but not originate them. If the remote device initiates a call, the MAX simply does not answer it. This situation could result in repeated charges for calls that have no purpose. To keep the remote device from originating calls, set AnsOrig=Ans Only for that device.

Dependencies: Keep this additional information in mind:

- The MAX adds or subtracts switched channels on a Nailed/Mpp connection as required by the settings on either side of the connection.

  Each side makes its calculations based on the traffic received at that side. If the two sides of the connection disagree on the number of channels needed, the side requesting the greater number prevails.
- The DO Hangup command works only from the caller side of the connection when you choose Nailed/Mpp.

Ascend-CBCP-Enable (112)

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

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

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: See Also: “Ascend-CBCP-Mode (113).”
“Ascend-CBCP-Trunk-Group (115).”

Ascend-CBCP-Mode (113)

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

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

Usage: 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: See Also: “Ascend-CBCP-Enable (112).”
“Ascend-CBCP-Trunk-Group (115).”

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 pre-appended to the caller-supplied number when the MAX calls back.

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

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 (112)."
"Ascend-CBCP-Mode (113)."

Ascend-Client-Assign-DNS (137)

Description: Specifies whether or not the MAX TNT sends the Ascend-Client-Primary-DNS and Ascend-Client-Secondary-DNS values during connection negotiation.

Usage: Specify one of the following settings:

- DNS-Assign-No (0) disables client DNS server negotiation for the link. DNS-Assign-No is the default.
- DNS-Assign-Yes (1) enables client DNS server negotiation for the link.

Dependencies: To direct the MAX TNT to send the client DNS server address during connection negotiation, you must include the setting Ascend-Client-Assign-DNS=DNS-Assign-Yes, and specify a valid DNS server by means of the Ascend-Client-Primary-DNS or Ascend-Cli-ent- Secondary-DNS attribute.

See Also: "Ascend-Client-Primary-DNS (135)."
"Ascend-Client-Secondary-DNS (136)."

Ascend-Client-Gateway (132)

Description: The Ascend-Client-Gateway attribute specifies the default route for IP packets coming from the user on this connection.

Usage: Specify the IP address of the next hop router in dotted decimal notation. The default value is 0.0.0.0. If you accept this value, the Ascend unit routes packets as specified in the routing table, using the system-wide default route if it cannot find a more specific route.

The Ascend unit must have a direct route to the address you specify. The direct route can take place via a profile or an Ethernet connection. If the Ascend unit does not have a direct route, it drops the packets on the connection. When you diagnose routing problems with a profile using this feature, an error in a per-user gateway address is not apparent from inspection of the global routing table.

Example: If you specify Ascend-Client-Gateway=10.0.0.3 in the RADIUS user profile Berkeley, IP packets from the user with destinations through the default route goes through the router at 10.0.0.3.

Ascend-Client-Primary-DNS (135)

Description: Specifies a primary DNS server address to send to any client connecting to the MAX TNT.

Usage: Specify the IP address of the primary DNS server. You must specify the address in dotted decimal notation. The default is 0.0.0.0, which specifies that no primary DNS server is available for the connection. If you do not specify Ascend-Client-Primary-DNS or Ascend-Client-Secondary-DNS in any user profile, the Ascend unit routes packets as specified in the routing table, using the system-wide default route if it cannot find a more specific route.

Dependencies: You must include the setting Ascend-Client-Assign-DNS=DNS-Assign-Yes to direct the MAX TNT to send the primary DNS server address during connection negotiation.

See Also: “Ascend-Client-Assign-DNS (137)”
“Ascend-Client-Secondary-DNS (136)”.

Ascend-Client-Secondary-DNS (136)

Description: Description: Specifies a secondary DNS server address to send to any client connecting to the MAX TNT.

Usage: Specify the IP address of the secondary DNS server. You must specify the address in dotted decimal notation. The default is 0.0.0.0, which specifies that no primary DNS server is available for the connection. If you do not specify Ascend-Client-Primary-DNS or Ascend-Client-Secondary-DNS in any user profile, the Ascend unit routes packets as specified in the routing table, using the system-wide default route if it cannot find a more specific route.

Dependencies: You must include the setting Ascend-Client-Assign-DNS=DNS-Assign-Yes to direct the MAX TNT to send the secondary DNS server address during connection negotiation.

See Also: "Ascend-Client-Assign-DNS (137)"
"Ascend-Client-Primary-DNS (135)".

Ascend-Connect-Progress (196)

Description: This attribute specifies the state of the connection before it is disconnected.

Ascend-Connect-Progress is included in an Accounting-Request packet when both of these conditions are true:

- The session has ended or has failed to authenticate (Acct-Status-Type=Stop).
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Ascend-Connect-Progress can have any one of values specified in Table C-6.

Table C-6. Ascend-Connect-Progress codes

Code	Explanation
0	No progress.
1	Not applicable.
2	The progress of the call is unknown.
10	The call is up.
30	The modem is up.
31	The modem is waiting for DCD.
32	The modem is waiting for result codes.
40	The terminal server session has started up.

Table C-6. Ascend-Connect-Progress codes

Code	Explanation
41	The TCP connection is being established.
42	The immediate Telnet connection is being established.
43	A raw TCP session has been established with the host. This code does not imply that the user has logged into the host.
44	An immediate Telnet connection has been established with the host. This code does not imply that the user has logged into the host.
45	The Rlogin session is being established.
46	An Rlogin session has been established with the host. This code does not imply that the user has logged into the host.
60	The LAN session is up.
61	LCP negotiations are allowed.
62	CCP negotiations are allowed.
63	IPNCP negotiations are allowed.
64	Bridging NCP negotiations are allowed.
65	LCP is in the Open state.
66	CCP is in the Open state.
67	IPNCP is in the Open state.
68	Bridging NCP is in the Open state.
69	LCP is in the Initial state.
70	LCP is in the Starting state.
71	LCP is in the Closed state.

Table C-6. Ascend-Connect-Progress codes

Code	Explanation
72	LCP is in the Stopped state.
73	LCP is in the Closing state.
74	LCP is in the Stopping state.
75	LCP is in the Request Sent state.
76	LCP is in the ACK Received state.
77	LCP is in the ACK Sent state.
80	IPXNCP is in the Open state.
90	V.110 is up.
91	V.110 is in the Open state.
92	V.110 is in the Carrier state.
93	V.110 is in the Reset state.
94	V.110 is in the Closed state.

Ascend-Data-Filter (242)

Description: Unlike the Filter Profiles in the MAX configuration interface, RADIUS filters are part of the outgoing or incoming RADIUS user profile. In other words, within any RADIUS users file defining a user profile, you can include values for Ascend-Data-Filter to define data filters for that profile. RADIUS filters are retrieved only when a call is placed using a RADIUS outgoing profile or answered using a RADIUS incoming profile.

Usage: Filter entries apply on a first-match basis. Therefore, the order in which filter entries are entered is significant.

If you make changes to a filter in a RADIUS user profile, the changes do not take effect until a call uses that profile.

IP data filter entries

Use this format for an IP data filter entry:

```
Ascend-Data-Filter="ip <dir> <action>
[dstip <dest ipaddr>\<subnet mask>][srcip <src ipaddr>\<sub-
net mask>]
[<proto> [dstport <cmp> <value>] [srcport <cmp> <value>]
[<est>]]"
```

Note: A filter definition cannot contain new lines. The syntax is shown on multiple lines here for printing purposes only.

Table C-7 describes each element of the syntax None of the keywords are case sensitive.

Table C-7. IP data filter syntax elements

Keyword or argument	**Description**
ip	The keyword “ip” indicates an IP filter.
<dir>	The <dir> argument indicates filter direction. You can specify “in” (to filter packets coming into the MAX) or “out” (to filter packets going out of the MAX).
<action>	<action> indicates what action the MAX should take with a packet that matches the filter. You can specify either “forward” or “drop”.
dstip <dest ipaddr>	“dstip” is a keyword indicating “destination IP address.” The filter applies to packets whose destination address matches the value of <dest ipaddr>. If a subnet mask portion of the address is present, the MAX compares only the masked bits. If you set <dest ipaddr> to 0.0.0.0, or if this keyword and its IP address specification are not present, the filter matches all IP packets.

Table C-7. IP data filter syntax elements

Keyword or argument	Description
srcip <src ipaddr>	"srcip" is a keyword indicating "source IP address." The filter applies to packets whose source address matches the value of <src ipaddr>. If a subnet mask portion of the address is present, the MAX compares only the masked bits. If you set <src ipaddr> to 0.0.0.0, or if this keyword and its IP address specification are not present, the filter matches all IP packets.
<proto>	<proto> indicates a protocol that you can specify as a name or a number. The filter applies to packets whose protocol field matches this value.The supported names and numbers are icmp (1), tcp (6), udp (17), and ospf (89). If you set <proto> to 0 (zero), the filter matches any protocol.
dstport <cmp> <value>	"dstport" is a keyword indicating "destination port." This argument is valid only when the protocol is tcp (6) or udp (17). If you do not specify a destination port, the filter matches any port. <cmp> is an argument indicating how to compare the specified value to the actual destination port. It can have the value <, =, >, or !=. <value> can be a number or a name. Supported names and numbers are ftp-data (20), ftp (21), telnet (23), smtp (25), nameserver (42), domain (53), tftp (69), gopher (70), finger (79), www (80), kerberos (88), hostname (101), nntp (119), ntp (123), exec (512), login (513), cmd (514), and talk (517).

Table C-7. IP data filter syntax elements

Keyword or argument	Description
srcport <cmp> <value>	“srcport” is a keyword indicating “source port.” It is valid only when the protocol is tcp (6) or udp (17). If you do not specify a source port, the filter matches any port. <cmp> is an argument indicating how to compare the specified value to the actual source port. It can have the value <, =, >, or !=. <value> can be a number or a name. Supported names and numbers are ftp-data (20), ftp (21), telnet (23), smtp (25), nameserver (42), domain (53), tftp (69), gopher (70), finger (79), www (80), kerberos (88), hostname (101), nntp (119), ntp (123), exec (512), login (513), cmd (514), and talk (517).
<est>	If you set this argument to 1, the filter matches a packet only if a TCP session is already established. It is valid only when the <proto> specification is tcp (6).

Generic data filter entries

Use this format for a generic data filter entry:

```
Ascend-Data-Filter="generic <dir> <action> <offset> <mask>
<value> <compare> [<more>]"
```

Note: A filter definition cannot contain new lines. The syntax is shown on multiple lines here for printing purposes only.

Table C-8 describes each element of the syntax. None of the keywords are case sensitive.

Table C-8. Generic data filter syntax elements

Keyword or argument	Description
generic	The keyword “generic” indicates a generic filter.

Table C-8. Generic data filter syntax elements

Keyword or argument	Description
<dir>	The <dir> argument indicates filter direction. You can specify "in" (to filter packets coming into the MAX) or "out" (to filter packets going out of the MAX).
<action>	<action> indicates what action the MAX should take with a packet that matches the filter. You can specify either "forward" or "drop".
<offset>	<offset> indicates the number of bytes masked from the start of the packet. The byte position specified by <offset> is called the byte-offset. Starting at the position specified by <offset>, the MAX applies the value of the <mask> argument. A mask hides the part of a number that appears behind the binary zeroes in the mask; for example, if you set <mask> to ffff0000 in hexadecimal format, the filter uses only the first 16 binary digits in the comparison, since f=1111 in binary format. The unit then compares the unmasked portion of the packet with the value specified by the <value> argument.
<mask>	This argument indicates which bits to compare in a segment of the packet; the mask cannot exceed 6 bytes (12 hexadecimal digits). A one-bit in the mask indicates a bit to compare; a zero-bit indicates a bit to ignore. The length of the mask specifies the length of the comparison.
<value>	This argument indicates the value to compare to the packet contents at the specified offset in the packet. The length of the value must be the same as the length of the mask; otherwise, the MAX ignores the filter.
<compare>	This argument indicates how the MAX compares a packet's contents to the value specified by <value>. You can specify == or !=, for Equal or NotEqual. The default value is Equal.

Table C-8. Generic data filter syntax elements

Keyword or argument	Description
<more>	If present, this argument specifies whether the MAX applies the next filter definition in the profile to the current packet before making the Forward or Drop decision. The <dir> and <action> values of the next entry must be the same as the <dir> and <action> of the current entry; otherwise, the MAX ignores the <more> flag.

Example: These are examples of IP data filter entries:

```
Ascend-Data-Filter="ip in drop"
```

```
Ascend-Data-Filter="ip out forward tcp"
```

```
Ascend-Data-Filter="ip out forward tcp dstip 10.0.200.3/16
srcip 10.0.200.25/16 dstport!=telnet"
```

```
Ascend-Data-Filter="ip out forward tcp dstip 10.0.200.3/16
srcip 10.0.200.25/16 icmp"
```

Note: A filter definition cannot contain new lines. The syntax is shown on multiple lines here for printing purposes only.

These are examples of generic data filter entries:

```
Ascend-Data-Filter="generic in drop 0 ffff 0080"
```

```
Ascend-Data-Filter="generic in drop 0 ffff != 0080 more"
```

```
Ascend-Data-Filter="generic in drop 16 ff aa"
```

Ascend-Data-Rate (Attribute197)

Description: This attribute specifies the data rate of the connection in bits per second.

Ascend-Data-Rate is included in an Accounting-Request packet when both of these conditions are true:

- The session has ended or has failed to authenticate (Acct-Status-Type=Stop).
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Ascend-Data-Rate does not appear in a user profile. Its default value is 0 (zero).

Ascend-Data-Svc (247)

Description: This attribute specifies the type of data service the link uses for outgoing calls.

Usage: The data service you specify must be available end-to-end. You can set the Ascend-Data-Svc attribute to one of the values listed in Table C-9

Table C-9. Ascend-Data-Svc settings

Setting	Description
Switched-Voice-Bearer (0)	This value applies only to calls made over an ISDN BRI or T1 PRI line. When you specify this setting, the MAX enables the network to place an end-to-end digital voice call for transporting data when a switched data service is not available.
Switched-56KR (1)	The call contains restricted data, guaranteeing that the data the MAX transmits meets the density restrictions of D4-framed TI lines. D4 specifies the D4 format, also known as the Superframe format, for framing data at the physical layer. This format consists of 12 consecutive frames, separated by framing bits. The call connects to the Switched-56 data service. The only services available to lines using inband signaling (T1 access lines containing one or more switched channels, and Switched-56 lines) are Switched-56K and Switched-56KR.
Switched-64K (2)	The call contains any type of data and connects to the Switched-64 data service.
Switched-64KR (3)	The call contains restricted data and connects to the Switched-64 data service.

Table C-9. Ascend-Data-Svc settings

Setting	Description
Switched-56K (4)	The call contains any type of data and connects to the Switched-56 data service. The only services available to lines using inband signaling (T1 access lines containing one or more switched channels, and Switched-56 lines) are Switched-56K and Switched- 56KR. For most T1 PRI lines, select Switched-56K.
Nailed-56KR (1)	The call contains restricted data and connects to the Nailed-56 data service.
Nailed-64K (2)	The call contains any type of data and connects to the Nailed-64 data service.

Dependencies: Keep this additional information in mind:

Determine the base bandwidth of a call by multiplying the value of the Ascend-Base-Channel-Count attribute by the value of the Ascend-Data-Svc attribute.

- Either party can request a data service that is unavailable; in this case, the MAX cannot connect the call.

Ascend-DBA-Monitor (171)

Description: This attribute species how the Ascend calling unit monitors the traffic on an MP+ call. The Ascend unit can use this information to add or subtract bandwidth as needed.

Usage: You can specify one of these values:

- DBA-Transmit (0)

  This setting indicates that the MAX adds or subtracts bandwidth based on the amount of data it transmits.

  Transmit is the default.
- DBA-Transmit-Recv (1)

  This setting indicates that the MAX adds or subtracts bandwidth based on the amount of data it transmits *and* receives.

- DBA-None (2)
  This setting indicates that the MAX does not monitor traffic over the link.

Dependencies: Keep this additional information in mind:

- Ascend-DBA-Monitor is supported only on MP+ calls.
- If both sides of the link have Ascend-DBA-Monitor set to None, Dynamic Bandwidth Allocation is disabled.

Ascend-Dec-Channel-Count (237)

Description: This attribute specifies the number of channels the MAX removes when bandwidth changes either manually or automatically during a call.

Usage: Specify a number between 1 and 32. The default value is 1.

Dependencies: Keep this additional information in mind:

- Ascend-Dec-Channel-Count does not apply if all channels of a link are nailed up
  (Ascend-Call-Type=Nailed).
- Ascend-Dec-Channel-Count applies only when the link is using MP+ encapsulation (Framed-Protocol=MPP).
- You cannot clear a call by decrementing channels.

Ascend-DHCP-Maximum-Leases (134)

Description: The Ascend-DHCP-Maximum-Leases attribute specifies the maximum number of dynamic addresses the MAX can assign to Network Address Translation (NAT) for LAN clients using this connection.

Usage: Specify a value between 1 and 254. The default is 4.

See Also: “Ascend-DHCP-Pool-Number (148)” “Ascend-DHCP-Reply (147)”

Ascend-DHCP-Pool-Number (148)

Description: The Ascend-DHCP-Pool-Number attribute indicates the address pool from which the MAX assigns a dynamic IP address to the Dynamic Host Configuration Protocol (DHCP) client.

Usage: Specify an integer between 1 and the number of address pools defined on the MAX The default value is 0 (zero), which specifies that the MAX uses the first defined IP address pool.

Dependencies: When the DHCP client requests an address, the MAX allocates an IP address from one of its IP address pools and assigns it to the client for 30 minutes. The client must renew the IP address assignment after the 30-minute period expires.

In its local memory, the MAX keeps track of all the IP addresses it has assigned. Therefore, it loses the entries for current, unexpired IP address assignments when you reset it. If a client holds an unexpired IP address assignment when you reset the MAX, the MAX may assign the same address to a new client. These duplicate IP addresses cause network problems until the first assignment expires or one of the clients reboots.

See Also: "Ascend-DHCP-Maximum-Leases (134)."
"Ascend-DHCP-Reply (147)."

Ascend-DHCP-Reply (147)

Description: The Ascend-DHCP-Reply attribute specifies whether the MAX processes DHCP packets and acts as a DHCP server on this connection.

Usage: You can specify one of these settings:

- DHCP-Reply-Yes indicates that the MAX processes DHCP packets.
- For a bridged connection, the MAX responds to all DHCP requests.
- For a non-bridged connection, the MAX responds only to Network Address Translation (NAT) for LAN DHCP packets.
- DHCP-Reply-No indicates that the MAX does not process DHCP packets, but routes or bridges DHCP packets as any other packet.
- The default value is DHCP-Reply-No.

See Also: "Ascend-DHCP-Maximum-Leases (134)." "Ascend-DHCP-Pool-Number (148.)"

Ascend-Dial-Number (227)

Description: This attribute specifies the phone number the MAX dials to reach the bridge, router, or node at the remote end of the link.

Usage: Specify a telephone number. You can enter up to 20 characters, and you must limit those characters to the following:

```
1234567890()[]!z-*#|
```

The MAX sends only the numeric characters to place a call. The default value is null.

If Use Trunk Grps=Yes in the System Profile, the first digits in the Ascend-Dial-Number attribute have the meanings listed in Table C-10.

Table C-10. Ascend-Dial-Number digits

Digit	Explanation
First digit is between 4 and 9.	The MAX places the call over the corresponding trunk group listed in the Ch *n* Trnk Grp, B1 Trnk Grp, or B2 Trnk Grp parameters in the Line Profile. If Dial Plan=Trunk Grp, the digits following the first digit constitute an ordinary phone number. If Dial Plan=Extended, the next two digits specify the Dial Plan Profile containing the parameters the MAX uses when making the call. These parameters constitute the extended dial plan. An ordinary phone number follows these two digits.
First digit is 3.	The MAX places the call to a destination listed in a Destination Profile. In this case, the second and third digits indicate the number of the Destination Profile.

Table C-10. Ascend-Dial-Number digits

Digit	Explanation
First digit is 2.	The MAX places the call between host ports on the same MAX, or between TEs (Terminal Equipment) on a local ISDN BRI line on the same MAX. The first type of call is a port-to-port call; the latter type of call is a TE-to-TE call. In a port-to-port call, the second digit indicates the slot of a serial host port module. In a TE-to-TE call, the second digit indicates the slot of a Host/BRI module. If you enter 0 (zero) for the second digit, the call connects to any available serial host port and ignores the third digit. If you enter a nonzero value for the second digit, the third digit selects the serial host port (for a port-to-port call) or a local ISDN BRI port (for a TE-to-TE call). If you enter 0 (zero) for the third digit, the call connects to any available serial host port or local ISDN BRI line in the module selected by the second digit.

Ascend-Dialout-Allowed (131)

Description: The Ascend-Dialout-Allowed attribute specifies whether the user associated with an outgoing RADIUS user profile can dial out using one of the MAX unit’s digital modems.

Usage: Usage: You can specify one of these settings:

- Dialout-Not-Allowed (0) indicates that the RADIUS user profile does not allow modem dialout.

  The default value is Dialout-Not Allowed.
- Dialout-Allowed (1) indicates that the RADIUS user profile allows modern dialout.

Ascend-Dsl-CIR-Xmit-Limit (101)

Description: Specifies the maximum data rate (in K-bits per second) to be transmitted across the connection. You can use this setting to limit bandwidth for a connection according to the rate charge for the account.

Usage: Specify a number from 0 to 64000. The default is 0 (zero), which disables the data-rate limit feature. If the value you specify is larger than the actual bandwidth provided by the line, the connection behaves as though the data rate limit were disabled, except that additional computations are performed unnecessarily.

Dependencies: The system activates configureable transmit data-rate limits only for connections that use CAP-RADSL, SDSL, and unchannelized DS3 cards. If you specify a value for a connection that does not use these cards, the system ignores the settings.

Ascend-Dsl-CIR-Recv-Limit (100)

Description: Specifies the maximum data rate (in K-bits per second) to be received across the connection. You can use this setting to limit bandwidth for a connection according to the rate charge for the account.

Usage: Specify a number from 0 to 64000. The default is 0 (zero), which disables the data-rate limit feature. If the value you specify is larger than the actual bandwidth provided by the line, the connection behaves as though the data rate limit were disabled, except that additional computations are performed unnecessarily.

Dependencies: The system activates configurable transmit data-rate limits only for connections that use CAP-RADSL, SDSL, and unchannelized DS3 cards. If you specify a value for a connection that does not use these cards, the system ignores the settings.

Ascend-Dsl-Downstream-Limit (99)

Description: Specifies the per-session ADSL-CAP downstream data rate.

Usage: Specify one of the following rates (in bps):

```
adslcap-dn-7168000 (0)
```

```
adslcap-dn-6272000 (1)
adslcap-dn-5120000 (2)
adslcap-dn-4480000 (3)
adslcap-dn-3200000 (4)
adslcap-dn-2688000 (5)
adslcap-dn-2560000 (6)
adslcap-dn-2240000 (7)
adslcap-dn-1920000 (8)
adslcap-dn-1600000 (9)
adslcap-dn-1280000 (10)
adslcap-dn-960000 (11)
adslcap-dn-640000 (12)
```

The default is adslcap-dn-2560000 (6).

Dependencies: “Ascend-Dsl-Rate-Mode (97)” on page 3-30 and “Ascend-Dsl-Rate-Type (92)” on page 3-30.

Ascend-Dsl-Rate-Mode (97)

Description: Specifies the per-session DSL data-rate mode.

Usage: At present, only the default Rate-Mode-AutoBaud setting is supported. This setting specifies that a DSL modem should train up to a set data rate. If a DSL modem cannot train to this data rate, it connects to the closest rate to which it can train (the modem’s ceiling rate).

See Also: “Ascend-DSL-Downstream-Limit (99)”
“Ascend-Dsl-Rate-Type (92)”.

Ascend-DSL-Rate-Type (92)

Description: Specifies the per-session modem type for rate control.

Usage: Specify one of the following settings:

- Rate-Type-Disabled (the default) specifies that modem rate control is not active for this connection.
- Rate-Type-AdslCap specifies that the per-session modem type is ADSL-CAP.

See Also: “Ascend-DSL-Downstream-Limit (99).”
“Ascend-Dsl-Rate-Mode (97).”

Ascend-Dsl-Upstream-Limit (98)

Description: Description: Specifies the per-session ADSL-CAP upstream data rate.

Usage: Usage: This release does not support setting the upstream rate.

See Also: See Also: “Ascend-DSL-Downstream-Limit (99)”.

Ascend-Discon-nect-Cause (195)

Description: This attribute specifies the reason a connection was taken off-line.

Ascend-Disconnect-Cause is included in an Accounting-Request packet when both of these conditions are true:

- The session has ended or has failed to authenticate (Acct-Status-Type=Stop).
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Ascend-Disconnect-Cause can return any of the values listed in Table C-11.

Table C-11. Ascend-Disconnect-Cause codes

Code	Description
0	No reason.
1	The event was not a disconnect.
2	The reason for the disconnect is unknown. This code can appear when the remote connection goes down.

Table C-11. Ascend-Disconnect-Cause codes

Code	Description
3	The call has been disconnected.
4	CLID authentication has failed.
These codes can appear if a disconnect occurs during the initial modem connection.	
10	The modem never detected DCD.
11	The modem detected DCD, but became inactive.
12	The result codes could not be parsed.
These codes are related to immediate Telnet and raw TCP disconnects during a terminal server session.	
20	The user exited normally from the terminal server.
21	The user exited from the terminal server because the idle timer expired.
22	The user exited normally from a Telnet session.
23	The user could not switch to SLIP or PPP because the remote host had no IP address or because the dynamic pool could not assign one.
24	The user exited normally from a raw TCP session.
25	The login process was terminated because the user failed to enter a correct password after three attempts.
26	The raw TCP option is not enabled.
27	The login process was terminated because the user typed Ctrl-C.
28	The terminal server session was terminated.
29	The user closed the virtual connection
30	The virtual connection was terminated.

Table C-11. Ascend-Disconnect-Cause codes

Code	Description
31	The user exited normally from an Rlogin session
32	The user selected an invalid Rlogin option.
33	The MAX has insufficient resources for the terminal server session.
These codes concern PPP connections.	
40	PPP LCP negotiation timed out while waiting for a response from a peer.
41	There was a failure to converge on PPP LCP negotiations.
42	PPP PAP authentication failed.
43	PPP CHAP authentication failed.
44	Authentication failed from the remote server.
45	The peer sent a PPP Terminate Request.
46	LCP got a close request from the upper layer while LCP was in an open state.
47	LCP closed because no NCPs were opened.
48	LCP closed because it could not determine to which MP bundle it should add the user.
49	LCP closed because no more channels could be added to an MP session.
These codes are related to immediate Telnet and raw TCP disconnects, and contain more specific information that the Telnet and TCP codes listed earlier in this table.	
50	The Raw TCP or Telnet internal session tables are full.
51	Internal resources are full.
52	The IP address for the Telnet host is invalid.

Table C-11. Ascend-Disconnect-Cause codes

Code	Description
53	The hostname could be resolved.
54	A bad or missing port number was detected.
These disconnect codes are returned by the TCP stack during an immediate Telnet or raw TCP session.	
60	The host reset the TCP connection.
61	The host refused the TCP connection.
62	The TCP connection timed out.
63	A foreign host closed the TCP connection.
64	The TCP network was unreachable.
65	The TCP host was unreachable.
66	The TCP network was administratively unreachable.
67	The TCP host was administratively unreachable.
68	The TCP port was unreachable.
These are additional disconnect codes.	
100	The session timed out because there was no activity on a PPP link.
101	The session failed for security reasons.
102	The session was terminated for callback.
120	The call was refused because the protocol was disabled or unsupported.
150	RADIUS requested the disconnect.
160	The allowed retries for V.110 synchronization have been exceeded.

Table C-11. Ascend-Disconnect-Cause codes

Code	Description
170	PPP authentication has timed out.

Ascend-Event-Type (150)

Description: Indicates one of the following:

- A cold-start notification, informing the accounting server that the MAX TNT has started up
- A session event, informing the authentication server that a session has begun

Usage: For a cold-start notification, Ascend-Event-Type=Ascend-Coldstart (1). For a session event, Ascend-Event-Type=Ascend-Session-Event (2).

Dependencies: In a cold-start notification, the MAX TNT sends values for NAS-Identifier, Ascend-Event-Type, and Ascend-Number-Sessions in an Ascend-Access-Event-Request packet (code 33). The RADIUS accounting server must send back an Ascend-Access-Event-Response packet (code 34) with the correct identifier to the MAX TNT.

In a session event, the MAX TNT sends values for Password, NAS-Identifier, Ascend-Access-Event-Type, and Ascend-Number-Sessions in an Ascend-Access-Event-Request packet (code 33). The authentication server must send back an Ascend-Access-Event-Response packet (code 34) with the correct identifier to the MAX TNT.

See Also: “Ascend-Number-Sessions (202).”

Ascend-Expect-Callback (149)

Description: The Ascend-Expect-Callback attribute specifies whether a user dialing out should expect the remote end to call back.

When the remote device is set to call back (Ascend-Callback=Callback-Yes or Callback=Yes) and CLID authentication is not required, the remote device answers the call, verifies a name and password against a user profile, hangs up, and dials back to the caller using these values:

- The phone number specified by Ascend-Dial-Number
- The password specified by Ascend-Send-Secret or Ascend-Send-Passwd
- Any other relevant attributes in the user profile that authenticated the call

If the remote RADIUS user profile is set up for callback, and the remote unit requires CLID-only authentication (Id Auth=Require), the remote device never answers the call. The caller can avoid billing charges. However, a problem can also occur. To the caller, it appears as though the call never got through at all. This is a special problem for Ping and Telnet, because these processes continuously try to open a connection and reject any callback.

When you set Ascend-Expect-Callback=Expect-Callback-Yes, calls that dial out and do not connect (for any reason) appear on a list that disallows any further calls to that destination for 90 seconds. This delay gives the remote device an opportunity to complete the callback.

Usage: You can specify one of these values:

- Expect-Callback-No (0) indicates that the caller does not wait for a callback after placing a call that does not connect.
- Expect-Callback-Yes (1) indicates that the caller waits 90 seconds after placing a call that does not connect before attempting to place another call to the same number.

See Also: “Ascend-Callback (246).”

Ascend-Filter (91)

Description: Specifies a string-format filter, which can include an IP TOS filter specification.

Usage: Filter entries apply on a first-match basis. Therefore, the order in which you enter them is significant. If you make changes to a filter in a RADIUS user profile, the changes do not take effect until a call uses that profile. A TOS filter value is specified in the following format:

```
iptos dir [dstip dest_ipaddr\subnet_mask]
[srcip src_ipaddr\subnet_mask][ proto][destport cmp
value]
```

```
[srcport cmp value][precedence value][type-of-service
value]
```

Note: A filter definition cannot contain new lines. The syntax is shown here on multiple lines for printing purposes only.

Table C-12. Keyword or argument description

Keyword or Argument	**Description**
ipto	Specifies an IP filter.
dir	Specifies filter direction. You can specify in (to filter packets coming into the MAX TNT) or out (to filter packets going out of the MAX TNT).
dstip dest_ipaddr \subnet_mask	If the dstip keyword is followed by a valid IP address, the TOS filter will set bytes only in packets with that destination address. If a subnet mask portion of the address is present, the MAX TNT compares only the masked bits. If the dstip keyword is followed by the zero address (0.0.0.0), or if this keyword and its IP address specification are not present, the filter matches all IP packets.
srcip src_ipaddr \subnet_mask	If the srcip keyword is followed by a valid IP address, the TOS filter will set bytes only in packets with that source address. If a subnet mask portion of the address is present, the MAX TNT compares only the masked bits. If the srcip keyword is followed by the zero address (0.0.0.0), or if this keyword and its IP address specification are not present, the filter matches all IP packets. proto A protocol number. A value of zero matches all protocols. If you specify a non-zero number, the MAX TNT compares it to the Protocol field in packets. For list of protocol numbers, see RFC 1700.
proto	A protocol number. A value of zero matches all protocols. If you specify a non-zero number, the MAX TNT compares it to the Protocol field in packets. For list of protocol numbers, see RFC 1700.

Table C-12. Keyword or argument description

Keyword or Argument	Description
dstport cmp value	If the dstport keyword is followed by a comparison symbol and a port, the port is compared to the destination port of a packet. The comparison symbol can be < (less-than), = (equal), > (greater-than), or != (not-equal). The port value can be one of the following names or numbers: ftp-data (20), ftp (21), telnet (23), smtp (25), nameserver (42), domain (53), tftp (69), gopher (70), finger (79), www (80), kerberos (88), hostname (101), nntp (119), ntp (123), exec (512), login (513), cmd (514), talk (517).
srcport cmp value	If the srcport keyword is followed by a comparison symbol and a port, the port is compared to the source port of a packet. The comparison symbol can be < (less-than), = (equal), > (greater-than), or != (not-equal). The port value can be one of the following names or numbers: ftp-data (20), ftp (21), telnet (23), smtp (25), nameserver (42), domain (53), tftp (69), gopher (70), finger (79), www (80), kerberos (88), hostname (101), nntp (119), ntp (123), exec (512), login (513), cmd (514), talk (517).

Table C-12. Keyword or argument description

Keyword or Argument	Description
precedence value	Specifies the priority level of the data stream. The three most significant bits of the TOS byte are priority bits used to set precedence for priority queuing. If a packet matches the filter, those bits are set to the specified value (most significant bit first): 000—Normal priority. 001—Priority level 1. 010—Priority level 2. 011—Priority level 3. 100—Priority level 4. 101—Priority level 5. 110—Priority level 6. 111—Priority level 7 (the highest priority).

Table C-12. Keyword or argument description

Keyword or Argument	Description
type-of-service value	Type of Service of the data stream. If a packet matches the filter, the system sets the four bits following the three most significant bits of the TOS byte to the specified value. Those four bits are used to choose a link based on the type of service. Specify one of the following values: Normal (0)—Normal service. Disabled (1)—Disables TOS. Cost (2)—Minimize monetary cost. Reliability (4)—Maximize reliability. Throughput (8)—Maximize throughput. Latency (16)—Minimize delay.

Example:

The following RADIUS user profile defines a TOS filter for TCP packets (protocol 6) that are destined for a single host at 10.168.6.24. The packets must be sent on TCP port 23. For incoming packets that match this filter, the priority is set at level 2. This is a relatively low priority, which means that an upstream router that implements priority queuing may drop these packets when it becomes loaded. The commands also set TOS to prefer a low latency connection. This means that the upstream router will choose a a fast connection is one is available, even if it is higher cost, lower bandwidth, or less reliable than another available link.

```
John Password="jlhkjtn", User-Service=Framed
   Framed-Protocol=PPP,
   Framed-IP-Address=10.168.6.120
   Framed-IP-Netmask=255.255.255.0
   Ascend-Filter="iptos in dstip 10.168.6.24/32
   dstport=23 precedence 010 type-of-service latency"
```

See Also: "Ascend-IP-TOS (88)."
"Ascend-IP-TOS-Apply-To (90)."
"Ascend-IP-TOS-Precedence (89)."

Ascend-First-Dest (189)

Description: This attribute records the destination IP address of the first packet received on a link after RADIUS authenticates the connection.

Ascend-First-Dest is included in an Accounting-Request packet when all of these conditions are met:

- The session has been authenticated.
- The session has ended (Acct-Status-Type=Stop).
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Ascend-First-Dest does not appear in a user profile and has no default value.

Dependencies: This attribute only applies if the session has been configured to route IP.

Ascend-Force-56 (248)

Description: This attribute specifies whether the MAX uses only the 56-kbps portion of a channel, even when all 64 kbps appear to be available:

Usage: You can specify one of these values:

- Force-56-No

  This setting indicates that the MAX should use the entire 64 kbps (when available). Force-56-No is the default.
- Force-56-Yes

This setting specifies that the MAX should use only the 56-kbps portion of a channel.

Set Ascend-Force-56=Force=56-Yes when you place calls to European or Pacific Rim countries from within North America and the complete path cannot distinguish between the Switched-56 and Switched-64 data services. This feature is not required if you are placing calls only within North America.

Ascend-FR-Circuit-Name (156)

Description: This attribute specifies the PVC (Permanent Virtual Connection) for which the user profile is an endpoint. A circuit specification defines two DLCI endpoints of a PVC, with one endpoint specified in each RADIUS user profile (or Connection Profile).

Usage: Specify a text string containing up to 15 characters. The default value is null.

Dependencies: Keep this additional information in mind:

- You can specify Ascend-FR-Circuit-Name only when Framed-Protocol=FR-CIR.
- You can specify Ascend-FR-Circuit-Name only for a gateway connection (when Ascend-FR-Direct=FR-Direct-No).
- Two profiles are required for a single PVC.

  You can use two RADIUS user profiles, two Connection Profiles, or one RADIUS user profile and one Connection Profile. The two DLCIs can use the same Frame Relay Profile or different ones.
- Pairs of links with matching Ascend-FR-Circuit-Name attributes (or Circuit parameters) are switched to each other; therefore, make sure that you specify the exact same name for Ascend-FR-Circuit-Name or the Circuit parameter in each profile.

Ascend-FR-DCE-N392 (162)

Description: This attribute specifies the number of errors during Ascend-FR-DCE-N393-monitored events that cause the network side to declare the user side's procedures inactive.

Usage: Specify an integer between 1 and 10. The default value is 3.

Dependencies: Keep this additional information in mind:

- Set Ascend-FR-DCE-N392 to a value less than Ascend-FR-DCE-N393.
- Ascend-FR-DCE-N392 does not apply if Ascend-FR-Type=Ascend-FR-DTE.

Ascend-FR-DCE-N393 (164)

Description: This attribute indicates the DCE-monitored event count. A link is always considered active if the value of Ascend-FR-DCE-N393 is not reached.

Usage: Specify a number between 1 and 10. The default value is 4.

Dependencies: This attribute does not apply if Ascend-FR-Type=Ascend-FR-DTE.

Ascend-FR-Direct (219)

Description: This attribute specifies whether the MAX uses a gateway connection or a redirect connection for frame relay packets.

Usage: You can specify one of these values:

- FR-Direct-No (0) indicates that the MAX uses a gateway connection.

  A gateway connection is a bridging or routing link between the MAX and a remote site via a frame relay switch. When the MAX receives IP packets destined for that site, it encapsulates the packets in frame relay (RFC 1490) and forwards the data stream out to the frame relay switch using the DLCI (Data Link Connection Indicator) specified by Ascend-FR-DLCI. The frame relay switch uses the DLCI to route the frames to the right destination.

  FR-Direct-No is the default.

- FR-Direct-Yes (1) indicates that the MAX uses a redirect connection.

  A redirect connection is designed only for forwarding incoming switched calls that use IP routing, such as regular PPP or MP+ calls. When the MAX receives IP packets from a caller that has a redirect specified in its RADIUS user profile, it simply forwards the data stream out to the frame relay switch using the DLCI (Data Link Connection Indicator) specified by Ascend-FR-Direct-DLCI. In so doing, the MAX effectively passes on the responsibility of routing those packets to a later hop on the frame relay network. The MAX never examines the destination address of redirect packets.

Ascend-FR-Direct-DLCI (221)

Description: This attribute specifies the DLCI (Data Link Connection Indicator) for the user profile in a frame relay redirect connection. The DLCI identifies the user profile to the frame relay switch as a logical link on a physical circuit.

Usage: Specify an integer between 16 and 991. The default value is 16. Many redirect connections can use the same DLCI.

Dependencies: Ascend-FR-Direct-DLCI applies only if Ascend-FR-Direct=FR-Direct-Yes.

Example: This portion of a user profile shows a redirect connection configured using DLCI 21 and the Frame Relay Profile called “Montgomery”.

```
permconn-max-1 Password="ascend", Service-Type=Outbound
    User-Name="Phani-gw-1",
   Ascend-FR-Direct=FR-Direct-Yes,
   Ascend-FR-Direct-Profile="Montgomery",
   Ascend-FR-Direct-DLCI=21,
   Metric=2,
   ...
```

Ascend-FR-Direct-Profile (220)

Description: This attribute specifies the name of the Frame Relay Profile that carries the redirect connection.

Usage: Indicate the name of a Frame Relay Profile that connects to the frame relay switch handling the DLCI (Data Link Connection Indicator) specified by

Ascend-FR-Direct-DLCI. You can specify up to 15 alphanumeric characters. The default value is null. Make sure that you enter the name exactly as it appears in the Name parameter of the Frame Relay Profile.

Dependencies: Ascend-FR-Direct-Profile applies only if Ascend-FR-Direct=FR-Direct-Yes.

Example: This portion of a user profile shows a redirect connection configured using DLCI 21 and the Frame Relay Profile called “Montgomery”.

```
permconn-max-1 Password="ascend", Service-Type=Outbound
   User-Name="Phani-gw-1",
  Ascend-FR-Direct=FR-Direct-Yes,
  Ascend-FR-Direct-Profile="Montgomery",
  Ascend-FR-Direct-DLCI=21,
  Metric=2,
  ...
```

Ascend-FR-DLCI (179)

Description: This attribute specifies the DLCI (Data Link Connection Indicator) for the user profile in a frame relay gateway connection. The DLCI identifies the user profile to the frame relay switch as a logical link on a physical circuit.

Usage: Specify an integer between 16 and 991. The default value is 16. You must assign each gateway connection its own DLCI.

Dependencies: Ascend-FR-DLCI applies only if Ascend-FR-Direct=FR-Direct-No.

Example: This portion of a user profile shows a gateway connection configured using DLCI 21 and the Frame Relay Profile called “Florence”.

```
permconn-max-1 Password="ascend", Service-Type=Outbound
   User-Name="Phani-gw-1",
  Ascend-FR-Direct=FR-Direct-No,
  Ascend-FR-Profile-Name="Florence",
  Ascend-FR-DLCI=21,
  Metric=2,
  ...
```

Ascend-FR-DTE-N392 (163)

Description: This attribute specifies the number of errors during Ascend-FR-DTE-N393-monitored events that cause the user side to declare the network side's procedures inactive.

Usage: Specify an integer between 1 and 10. The default value is 3.

Dependencies: Keep this additional information in mind:

- Set Ascend-FR-DTE-N392 to a value less than Ascend-FR-DTE-N393.
- Ascend-FR-DTE-N392 does not apply if Ascend-FR-Type=Ascend-FR-DCE.

Ascend-FR-DTE-N393 (165)

Description: This attribute indicates the DTE-monitored event count. A link is always considered active if the value of Ascend-FR-DTE-N393 is not reached.

Usage: Specify a number between 1 and 10. The default value is 4.

Dependencies: This attribute does not apply if Ascend-FR-Type=Ascend-FR-DCE.

Ascend-FR-Link-Mgt (160)

Description: In a Frame Relay Profile, this attribute specifies the link management protocol used between the MAX and the frame relay switch.

Usage: You can specify one of these values:

- Ascend-FR-No-Link-Mgt (0)

  This setting indicates no link management, and is the default. A link is always considered active if no link management functions are performed.
- Ascend-FR-T1-617D (1)

  This setting indicates T1.617 Annex D link management.
- Ascend-FR-Q-933A (2)

  This setting indicates Q.933 Annex A link management.

Ascend-FR-Link-Status-DLCI (106)

Description: Specifies the DLCI to use for link management on the Frame Relay datalink.

Usage: Accept the default of 0 (zero) or specify DLCI 1023.

See Also: "Ascend-FR-Link-Mgt (160)."

Ascend-FR-LinkUp (157)

Description: In a Frame Relay Profile, this attribute specifies whether the frame relay link comes up automatically.

Usage: You can specify one of these values:

- Ascend-LinkUp-Default (0)

  This setting indicates that the datalink does not come up unless a DLCI brings it up, and shuts down after the last DLCI has been removed. This value is the default.
- Ascend-LinkUp-AlwaysUp (1)

  This setting indicates that the datalink comes up automatically and stays up even when the last DLCI has been removed.

Dependencies: You can start and drop frame relay connections by using the DO DIAL and DO HANGUP commands. DO DIAL brings up a connection. DO HANGUP closes the link and any DLCIs on it. If Ascend-FR-LinkUp=Ascend-LinkUp-AlwaysUp, DO HANGUP brings the link down, but the link automatically restarts. A restart also occurs if a DLCI brings up the datalink.

Ascend-FR-N391 (161)

Description: In a Frame Relay Profile, this attribute specifies the interval in seconds at which the MAX requests a Full Status Report.

If the frame relay link is configured for link management, it regularly request updates on the status of the link. The frame relay unit at the other end of the link must respond to these requests; otherwise, the MAX considers the link inactive. Furthermore, if the response to these requests indicates a DLCI failure, the MAX considers the link inactive.

Usage: Specify an integer between 1 and 255. The default value is 6.

Dependencies: This attribute does not apply if Ascend-FR-Type=Ascend-FR-DCE.

Ascend-FR-Nailed-Grp (158)

Description: This attribute associates a group of nailed-up channels with the Frame Relay Profile.

Usage: Specify a number between 1 and the maximum number of nailed-up channels that your MAX allows. The default value is 1.

Dependencies: Do not associated a group with more than one active Frame Relay Profile.

Ascend-FR-Profile-Name (180)

Description: This attribute specifies the name of the Frame Relay Profile that carries the gateway connection.

Usage: Indicate the name of a Frame Relay Profile that connects to the frame relay switch handling the DLCI (Data Link Connection Indicator) specified by Ascend-FR-DLCI. You can specify up to 15 alphanumeric characters. The default value is null. Make sure that you enter the name exactly as it appears in the Name parameter of the Frame Relay Profile.

Dependencies: Ascend-FR-Profile-Name applies only if Ascend-FR-Direct=FR-Direct-No.

Example: This portion of a user profile shows a gateway connection configured using DLCI 21 and the Frame Relay Profile called “Florence”.

```
permconn-max-1 Password="ascend", Service-Type=Outbound
     User-Name="Phani-gw-1",
    Ascend-FR-Direct=FR-Direct-No,
    Ascend-FR-Profile-Name="Florence",
    Ascend-FR-DLCI=21,
    Metric=2,
    ...
```

Ascend-FR-T391 (166)

Description: This attribute indicates the Link Integrity Verification polling timer.

Usage: You can specify a number of seconds between 5 and 30. The default value is 10.

Dependencies: This attribute does not apply if Ascend-FR-Type=Ascend-FR-DCE.

Ascend-FR-T392 (167)

Description: This attribute indicates the timer for the verification of the polling cycle— the length of time the unit should wait between Status Enquiry messages. An error is recorded if no Status Enquiry is received within the number seconds specified by this attribute.

Usage: Specify a number of seconds between 5 and 30. The default value is 10.

Dependencies: This attribute does not apply if Ascend-FR-Type=Ascend-FR-DTE.

Ascend-FR-Type (159)

Description: This attribute specifies the type of frame relay connection used by the Frame Relay Profile.

You can specify one of these values:

- Ascend-FR-DTE (0)

  This setting indicates a UNI-DTE interface. This value is the default. When you specify this value, the MAX performs DTE functions for link management, and can connect to a frame relay DCE unit—a frame relay switch.

  Choose this setting when Framed-Protocol=FR in a user profile for a gateway connection.
- Ascend-FR-DCE (1)

  This setting indicates a UNI-DCE interface. When you specify this value, the MAX performs DCE functions for link management, and can connect to a frame relay DTE unit—the user’s CPE (Customer Premises Equipment).

Choose this setting when Framed-Protocol=FR-CIR in a user profile for a gateway connection.

- Ascend-FR-NNI (2)

  This setting indicates an NNI interface. When you specify this value, the MAX performs both DTE and DCE functions for link management, and can connect to another NNI unit.

  Choose this setting when Framed-Protocol=FR-CIR in a user profile for a gateway connection.

Dependencies: Ascend-FR-Type is applicable only when a frame relay user profile specifies a gateway connection (Ascend-FR-Direct=FR-Direct-No) and Framed-Protocol=FR or FR-CIR.

Ascend-FT1-Caller (175)

Description: This attribute specifies whether the MAX initiates an FT1-AIM or an
FT1-B&O call, or whether it waits for the remote end to initiate these types of calls.

Usage: You can specify one of these values:

- FT1-No (0) specifies that the MAX waits for the remote end to initiate the call.

  FT1-No is the default.

- FT1-Yes (1) specifies that the MAX initiates the call.

  If you choose this setting, the MAX dials to bring on-line any switched circuits that are part of the call.

Dependencies: Keep this additional information in mind:

- If the remote end has set the Ascend-FT1-Caller attribute to FT1-No (or set the FT1 Caller parameter to No), set Ascend-FT1-Caller to FT1-Yes for the local MAX.
- If the remote end has set the Ascend-FT1-Caller attribute to FT1-Yes (or set the FT1 Caller parameter to Yes), set Ascend-FT1-Caller to FT1-No for the local MAX.

Ascend-Group (178)

Description: This attribute points to the nailed-up channels used by the profile's WAN link.

If you set the Ascend-Group attribute to a value that matches the settings of a Ch *n* Prt/Grp, B1 Prt/Grp, or B2 Prt/Grp parameter in a Line Profile, the MAX uses the specified channels for this profile's link across the WAN. Similarly, if Ascend-Group has the same value as Nailed Grp in the Serial WAN Profile, the MAX uses the serial WAN circuit for this profile's link.

Usage: Your usage depends upon the value you specify for the Ascend-Call-Type attribute:

- If you set Ascend-Call-Type to Nailed, you can specify a number between 1 and 60 for Ascend-Group.

  The default value is 1.
- If you set Ascend-Call-Type to Nailed/Mpp, you can use the Ascend-Group attribute to assign multiple nailed-up groups to the profile.

  Specify a single number, or specify a list of numbers between 1 and 60, separated by commas. Do not include spaces. The default value is 1.

Dependencies: Keep this additional information in mind:

- The Ascend-Group attribute does not apply if the link consists entirely of switched channels.
- If you add channels for the Ascend-Group attribute, the MAX adds the additional channels to any on-line connection that uses the group.
- Do not duplicate group numbers in active profiles—that is, choose a value for Ascend-Group that is not used by any other active Connection Profile, Call Profile, Frame Relay Profile, or RADIUS user profile.
- Although you can assign multiple groups to a user profile, do not mix the Serial WAN circuit with nailed-up BRI or T1/E1 channels.

Example: If Ascend-Call-Type=Nailed/Mpp, setting the Ascend-Group attribute to "1,3,5,7" assigns four nailed-up groups to the profile.

Ascend-Handle-IPX (222)

Description: This attribute specifies how the MAX handles NCP watchdog requests on behalf of IPX clients during IPX bridging.

Usage: You can specify one of these values:

- Handle-IPX-None (0)

  This setting indicates that special IPX behavior does not take place. Choose this setting when the LAN on each side of the bridge has one or more IPX servers.

  Handle-IPX-None is the default.

- Handle-IPX-Client (1)

  This setting indicates that the MAX discards RIP (Routing Information Protocol) and SAP
  (Service Advertising Protocol) periodic broadcasts at its WAN interface, but forwards RIP and SAP queries.

  The WAN interface is the port on the MAX that is connected to a WAN line. RIP and SAP queries enable a client workstation to locate a NetWare server across the network. Choose this setting when both these conditions are true:

  - The local LAN has IPX clients but no servers.
  - The MAX is acting as a bridge to another LAN containing only IPX servers or a combination of IPX servers and clients.

- Handle-IPX-Server (2)

  This setting indicates that the MAX discards all RIP (Routing Information Protocol) and SAP (Service Advertising Protocol) periodic broadcasts and queries at its WAN interface.

  This mode enables the MAX to bring down calls during idle periods without breaking
  client/server or peer-to-peer connections.

  Ordinarily, when a NetWare server does not receive a reply to the watchdog session
  keep alive packets it sends to a client, it closes the connection. When you specify Handle-IPX-Server, however, the MAX replies to NCP watchdog requests on behalf of clients on the other side of the bridge; in other words, the MAX tricks the server watchdog process into believing that the link is still active. This process is called watchdog spoofing.

  Choose this setting when both these conditions are true:

- The MAX is acting as a bridge to a remote LAN with IPX clients, but no servers.
- The local LAN contains only IPX servers, or a combination of IPX clients and
  servers.

Dependencies: Keep this additional information in mind:

- If you specify Ascend-Handle-IPX=Handle-IPX-Server, you must also specify a value for the Ascend-Netware-timeout attribute, indicating the maximum length of idle time during which the MAX performs watchdog spoofing for NetWare connections.
- If the connection does not bridge (Ascend-Bridge=Bridge-No), the Ascend-Handle-IPX attribute does not apply.
- If the MAX on one LAN sets Ascend-Handle-IPX=Handle-IPX-Server and the LAN on the other side of the connection has only NetWare clients, the MAX on the client-only LAN should set Ascend-Handle IPX=Handle-IPX-Client.

  If both LANs contain servers, both sides of the connection should set Ascend-Handle- IPX=Handle-IPX-None.
- Although Ascend-Handle-IPX does not apply if Ascend-Bridge=Bridge-No, the MAX automatically performs watchdog spoofing just as though you had set Ascend-Handle-IPX=Handle-IPX-Server; however, the MAX does not filter as though you had set Ascend-Handle-IPX=Handle-IPX-Server.

Example: This user profile specifies an IPX bridging link in which the local Ethernet supports NetWare clients, and the remote network supports both NetWare servers and clients:

```
MAX1 Password="m2dan", Service-Type=Framed
     Framed-Protocol=PPP,
     Ascend-Route-IPX=Route-IPX-No,
     Ascend-Bridge=Bridge-Yes,
     Ascend-Handle-IPX=Handle-IPX-Client,
     Ascend-Netware-timeout=30
```

Ascend-History-Weigh-Type (239)

Description: This attribute specifies which Dynamic Bandwidth Allocation (DBA) algorithm to use for calculating average line utilization (ALU) of transmitted data. DBA enables you to specify that the MAX uses ALU as the basis for automatically adding or subtracting bandwidth from a switched connection without terminating the link.

Usage: Figure C-1 illustrates the differences between the algorithms you can choose.

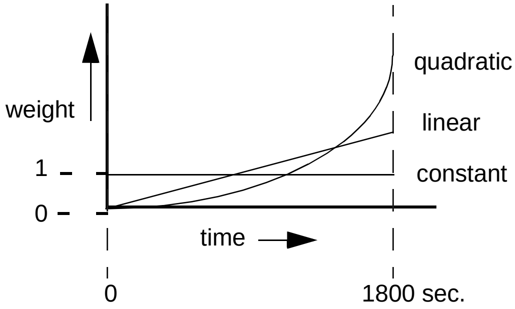


Figure C-1. Bandwidth algorithms for MP+ calls

- History-Constant (0) gives equal weight to all samples taken during the historical time period specified by the Ascend-Seconds-Of History attribute.

  When you select this option, older historical samples have as much impact on the decision to change bandwidth allocation as do more recent samples.

- History-Linear (1) gives more weight to recent samples of bandwidth usage than to older samples taken during the historical period specified by Ascend-Seconds-Of-History.

  The weighting grows at a linear rate.

- History-Quadratic (2) gives more weight to recent samples of bandwidth usage than to older samples taken during the historical period specified by the Ascend-Seconds-Of-History attribute.

  The weighting grows at a quadratic rate. History-Quadratic is the default.

Ascend-Home-Agent-IP-Addr (183)

Description: In a mobile node's RADIUS user profile, this attribute indicates the IP address of the home agent under ATMP (Ascend Tunnel Management Protocol) operation.

The RADIUS server passes the attributes contained in the mobile node's RADIUS user profile to the foreign agent; the foreign agent sends these attributes when connecting with the home agent.

A mobile node can also connect directly to the home agent. An ATMP-based RADIUS entry that is local to the home agent enables the mobile node to bypass a foreign agent connection, but does not preclude a foreign agent. If both the home agent and the foreign agent have local RADIUS entries for the mobile node, the node can choose between a direct connection or a tunneled connection through the foreign agent.

Usage: Specify an IP address in dotted decimal notation. An IP address consists of four numbers between 0 and 255, separated by periods. The default value is 0.0.0.0.

Example: The following RADIUS entry authenticates a mobile NetWare client that connects directly to the home agent. In this example, the home agent is configured in gateway mode; it forwards packets from the mobile node across a nailed WAN link to the home IPX network.

```
mobile-ipx Password="unit"
        Service-Type=Framed,
        Ascend-Route-IPX=Route-IPX-Yes,
        Framed-Protocol=PPP,
        Ascend-IPX-Peer-Mode=IPX-Peer-Dialin,
        Framed-IPX-Network=40000000,
        Ascend-IPX-Node-Addr=12345678,
        Ascend-Home-Agent-IP-Addr=200.168.6.18,
        Ascend-Home-Network-Name="dave's max",
        Ascend-Home-Agent-Password="pipeline"
```

Ascend-Home-Agent-Password (184)

Description: In a mobile node's RADIUS user profile, this attribute specifies the password that the foreign agent sends to the home agent in order to authenticate itself during ATMP (Ascend Tunnel Management Protocol) operation. This password must match the value of the Password parameter in the ATMP configuration in the Ethernet Profile for the home agent. All mobile nodes accessing a single home agent must specify the same password.

The RADIUS server passes the attributes contained in the mobile node's RADIUS user profile to the foreign agent; the foreign agent sends these attributes when connecting with the home agent.

A mobile node can also connect directly to the home agent. An ATMP-based RADIUS entry that is local to the home agent enables the mobile node to bypass a foreign agent connection, but does not preclude a foreign agent. If both the home agent and the foreign agent have local RADIUS entries for the mobile node, the node can choose between a direct connection or a tunneled connection through the foreign agent.

Usage: Specify a text string containing up to 20 characters. The default value is null.

Example: The following RADIUS entry authenticates a mobile NetWare client that connects directly to the home agent. In this example, the home agent is configured in gateway mode; it forwards packets from the mobile node across a nailed WAN link to the home IPX network.

```
mobile-ipx Password="unit"
        Service-Type=Framed,
        Ascend-Route-IPX=Route-IPX-Yes,
        Framed-Protocol=PPP,
        Ascend-IPX-Peer-Mode=IPX-Peer-Dialin,
        Framed-IPX-Network=40000000,
        Ascend-IPX-Node-Addr=12345678,
        Ascend-Home-Agent-IP-Addr=200.168.6.18,
        Ascend-Home-Network-Name="dave's max",
        Ascend-Home-Agent-Password="pipeline"
```

Ascend-Home-Agent-UDP-Port (186)

Description: In a mobile node's RADIUS user profile, this attribute specifies the UDP port number on the home agent to which the foreign agent directs ATMP (Ascend Tunnel Management Protocol) messages.

Usage: Specify a UDP port number between 0 and 65535. The default value is 5150.

Ascend-Home-Network-Name (185)

Description: In a mobile node's RADIUS user profile, this attribute specifies the name of the Connection Profile on which the home agent sends all packets it receives from the mobile node during ATMP (Ascend Tunnel Management Protocol) operation.

The RADIUS server passes the attributes contained in the mobile node's RADIUS user profile to the foreign agent; the foreign agent sends these attributes when connecting with the home agent.

A mobile node can also connect directly to the home agent. An ATMP-based RADIUS entry that is local to the home agent enables the mobile node to bypass a foreign agent connection, but does not preclude a foreign agent. If both the home agent and the foreign agent have local RADIUS entries for the mobile node, the node can choose between a direct connection or a tunneled connection through the foreign agent.

Usage: Specify the name of the home agent's Connection Profile. The default value is null.

Dependencies: You must specify a value for this attribute only if the home agent is a gateway (that is, only if Type=Gateway in the ATMP configuration for the Ethernet Profile).

Example: The following RADIUS entry authenticates a mobile NetWare client that connects directly to the home agent. In this example, the home agent is configured in gateway mode; it forwards packets from the mobile node across a nailed WAN link to the home IPX network.

```
mobile-ipx Password="unit"
       Service-Type=Framed,
       Ascend-Route-IPX=Route-IPX-Yes,
       Framed-Protocol=PPP,
       Ascend-IPX-Peer-Mode=IPX-Peer-Dialin,
       Framed-IPX-Network=40000000,
       Ascend-IPX-Node-Addr=12345678,
       Ascend-Home-Agent-IP-Addr=200.168.6.18,
       Ascend-Home-Network-Name="dave's max",
       Ascend-Home-Agent-Password="pipeline"
```

Ascend-Host-Info (252)

Description: This attribute specifies a list of hosts to which a user can establish a Telnet session.

Usage: You can specify up to 10 Ascend-Host-Info entries in a user profile. Enter your attribute settings in this format:

```
Ascend-Host-Info="<IP_address> <text>"
```

- <IP_address> specifies the IP address of each host.

  Specify an IP address in dotted decimal notation. An IP address consists of four numbers between 0 and 255, separated by periods. The default value is 0.0.0.0.

- <text> describes each host.

  You can enter up to 31 characters for <text>. The default value is null. The RADIUS server assigns the text a number; when the user selects the number, the terminal server initiates a Telnet session with the host at the specified IP address.

Dependencies: If you specify a value for the Ascend-Host-Info attribute, you must also make these settings in the TServ Options menu of the Ethernet Profile:

- Set Initial Scrn=Menu or Toggle Scrn=Yes.
- Set Remote Conf=Yes.

Example: Here is an example for a MAX named Cal:

```
initial-banner-Cal Password="ascend", Service-Type=Outbound
     Reply-Message="Up to 16 lines of up to 80 characters
     each",
     Reply-Message="will be accepted. Long lines will be
     truncated",
     Reply-Message="Additional lines will be ignored.",
     Reply-Message="",
     Ascend-Host-Info="1.2.3.4 Berkeley",
     Ascend-Host-Info="1.2.3.5 Alameda",
     Ascend-Host-Info="1.2.36 San Francisco",
     ...
```

Ascend-Idle-Limit (244)

Description: This attribute specifies the number of seconds the MAX waits before clearing a call when a session is inactive.

Usage: Specify a number between 0 and 65535. If you specify 0 (zero), the MAX always clears a call when a session is inactive. The default value is 120 seconds. If you accept the default and an existing Answer Profile specifies a value for the analogous Idle parameter, the Idle value is ignored and the MAX uses the Ascend-Idle-Limit default.

Dependencies: Keep this additional information in mind:

- If the time set by the Ascend-Idle-Limit expires, the call disconnects whether or not bandwidth utilization falls below the Ascend-MPP-Idle-Percent setting.
- When bandwidth utilization falls below the Ascend-MPP-Idle-Percent setting, the call disconnects regardless of whether the time specified by the Ascend-Idle-Limit attribute has expired.
- Because the Ascend-MPP-Idle-Percent attribute is dependent on traffic levels on both sides of the connection, we recommend that you use the Ascend-Idle-Limit attribute in preference to it.
- The Ascend-Idle-Limit attribute does not apply to nailed-up links.

Ascend-IF-Netmask (154)

Description: This attribute specifies the subnet mask in use for the local numbered interface.

Usage: Specify a subnet mask consisting of four numbers between 0 and 255, separated by periods. The default value is 0.0.0.0.

Ascend-Inc-Channel-Count (236)

Description: This attribute specifies the number of channels the MAX adds when bandwidth changes either manually or automatically during a call.

Usage: Specify a number between 1 and 32. The default value is 1.

Dependencies: Keep this additional information in mind:

- Ascend-Inc-Channel-Count does not apply if all channels of a link are nailed up (Ascend-Call-Type=Nailed).
- Ascend-Inc-Channel-Count applies only if the link is using MP+ encapsulation (Framed-Protocol=MPP).
- MP+ calls cannot exceed 32 channels.
- The sum of Ascend-Base-Channel-Count and Ascend-Inc-Channel-Count cannot exceed the maximum number of channels available.

Ascend-IP-Direct (209)

Description: This attribute specifies the IP address to which the MAX redirects packets from the user. When you include this attribute in a user profile, the MAX bypasses all internal routing and bridging tables, and simply sends all packets received on this connection’s WAN interface to the specified IP address.

Ascend-IP-Direct does not affect packets sent to this connection. Traffic destined for the connection user is routed using the MAX unit’s routing scheme.

Usage: Specify an IP address in dotted decimal notation. An IP address consists of four numbers between 0 and 255, separated by periods. The default value is 0.0.0.0. If you accept the default, the MAX does not redirect IP traffic.

Dependencies: Keep this additional information in mind:

- You can specify the Ascend-IP-Direct attribute only under these conditions:
  - IP routing is in use.
  - The user profile contains the specification Ascend-Bridge=Bridge-No.
  - Framed-Protocol is not set to COMB or FR.
- Do not set Ascend-IP-Direct and Ascend-FR-Direct in the same user profile; if you do, an error occurs.

- Ascend-IP-Direct connections typically turn off RIP.

  If the connection is configured to receive RIP, all RIP packets from the remote end are kept locally and forwarded to the IP address you specify. To turn off RIP, set Framed-Routing=None.

Example: This user profile specifies that the MAX redirects incoming packets to the host at IP address 10.2.3.11:

```
emma Password="m2dan", Service-Type=Framed
     Framed-Protocol=PPP,
     Framed-IP-Address=10.8.9.10,
     Framed-IP-Netmask=255.255.252.0,
     Ascend-Route-IP=Route-IP-Yes,
     Ascend-Bridge=Bridge-No,
     Ascend-IP-Direct=10.2.3.11,
     Ascend-Metric=2,
     Framed-Routing=None,
     ...
```

Ascend-IP-Pool-Definition (217)

Description: This attribute specifies the first IP address in an IP address pool, and indicates the number of addresses in the pool.

Usage: The Ascend-IP-Pool-Definition attribute has this format:

```
Ascend-IP-Pool-Definition="<num> <first_ipaddr>
<max_entries>"
```

Table C-13 describes each Ascend-IP-Pool-Definition argument.

Table C-13. Ascend-IP-Pool-Definition arguments

Argument	Description
<num>	Indicates the number of the pool. The default value is 1. Specify pool numbers starting with 1, unless you have defined pools in the MAX interface using the Pool #1 Start, Pool #1 Count, Pool #2 Start, and Pool #2 Count parameters and do not wish to override these settings. In this case, specify 3 for the first pool number in the RADIUS pseudo-user entry.
<first_ipaddr>	Specifies the first IP address in the address pool. The address you indicate should not accept a subnet mask, because it always becomes a host route. The default value is 0.0.0.0.
<max_entries>	Specifies the maximum number of IP addresses in the pool. Addresses are assigned sequentially, from <first_ipaddr> on, up to the limit of addresses specified by <max_entries>. The default value is 0.

Dependencies: You specify one or more Ascend-IP-Pool-Definition attributes in a pseudo-user profile. You create a pseudo-user to store information that the MAX can query—in this case, in order to store IP address pool information. Specify the first line of a pseudo-user entry in this format:

```
pools-<unit_name> Password="ascend", Service-Type=Outbound
```

<unit_name> is the system name of the MAX—that is, the name specified by the Name parameter in the System Profile. On the next lines of the profile, specify one or more Ascend-IP-Pool-Definition attributes.

Example: In this example, two IP address pools are created for the MAX to use. Address pool #1 contains a block of 7 IP addresses from 10.1.0.1 to 10.1.0.7. Address pool #2 contains a block of 48 IP addresses from 10.2.0.1 to 10.2.0.48.

```
pools-MAX Password="ascend", Service-Type=Outbound
    Ascend-IP-Pool-Definition="1 10.1.0.1 7",
    Ascend-IP-Pool-Definition="2 10.2.0.1 48"
```

Ascend-IP-TOS (88)

Description: Specifies the Type-of-Service (TOS) of the data stream.

Usage: The value you specify sets the four bits following the three most significant bits of the TOS byte. which are used to choose a link based on the type of service. Specify one of the following values:

- Ascend-IP-TOS IP-TOS-Normal (0) specifies normal service.
- Ascend-IP-TOS IP-TOS-Disabled (1) disables TOS.
- Ascend-IP-TOS IP-TOS-Cost (2) minimizes monetary cost.
- Ascend-IP-TOS IP-TOS-Reliability (4) maximizes reliability.
- Ascend-IP-TOS IP-TOS-Throughput (8) maximizes throughput.
- Ascend-IP-TOS IP-TOS-Latency (16) minimizes delay.

See Also: "Ascend-IP-TOS-Apply-To (90)."
"Ascend-IP-TOS-Precedence (89)."

Ascend-IP-TOS-Apply-To (90)

Description: Specifies the direction in which Type-of-Service (TOS) is enabled.

Usage: Specify one of the following values:

- IP-TOS-Apply-To-Incoming (1024) specifies that bits are set in packets received on the interface. This setting is the default.
- IP-TOS-Apply-To-Outgoing (2048) specifies that bits are set in outbound packets only.
- P-TOS-Apply-To-Both (3072) specifies that both incoming and outgoing packets are tagged.

See Also: : "Ascend-IP-TOS (88)."
"Ascend-IP-TOS-Precedence (89)."

Ascend-IP-TOS-Precedence (89)

Description: Specifies the priority level of the data stream.

Usage: The three most significant bits of the TOS byte are priority bits used to set precedence for priority queuing. When TOS is enabled, those bits can be set to one of the following values (most significant bit first):

- IP-TOS-Precedence-Pri-Normal (0) specifies normal priority.
- IP-TOS-Precedence-Pri-One (32) specifies priority level 1.
- IP-TOS-Precedence-Pri-Two (64) specifies priority level 2.
- IP-TOS-Precedence-Pri-Three (96) specifies priority level 3.
- IP-TOS-Precedence-Pri-Four (128) specifies priority level 4.
- IP-TOS-Precedence-Pri-Five (160) specifies priority level 5.
- IP-TOS-Precedence-Pri-Six (192) specifies priority level 6.
- IP-TOS-Precedence-Pri-Seven (224) specifies priority level 7 (the highest priority).

See Also: "Ascend-IP-TOS (88)."
"Ascend-IP-TOS-Apply-To (90)."

Ascend-IPX-Alias (224)

Description: This attribute specifies an IPX network number to use when connecting to IPX routers that require numbered interfaces.

Usage: Specify an IPX network number. The default value is 0 (zero). RADIUS requires that this attribute have a decimal value (base 10), but IPX network numbers generally have hexadecimal values (base 16). In order to give this attribute a value, you must convert the hexadecimal IPX network number to a decimal value for use in the user profile.

Ascend-IPX-Node-Addr (182)

Description: This attribute specifies a unique IPX node address on the network specified by Framed-IPX-Network. This value completes the IPX address of a mobile node.

Usage: Specify a 12-digit ASCII string enclosed in double-quotes. The RADIUS server passes the attributes contained in the mobile node's profile to the foreign agent; the foreign agent sends these attributes when connecting with the home agent.

Ascend-IPX-Peer-Mode (216)

Description: This attribute specifies whether the caller associated with the user profile is an Ethernet client with its own IPX network address, or a dial-in PPP client.

Dial-in clients do not belong to an IPX network, so they must be assigned an IPX network number to establish a routing connection with the MAX. To provide an IPX network number for dial-in clients, you must define a "virtual" IPX network in the Ethernet Profile using the IPX Pool# parameter. The MAX advertises the route to this virtual network and assigns it as the network address for dial-in clients.

Usage: For the Ascend-IPX-Peer-Mode attribute, you can specify one of these values:

- IPX-Peer-Router (0) indicates that the caller is on the Ethernet network and has its own IPX address.

  IPX-Peer-Router is the default.
- IPX-Peer-Dialin (1) indicates that the caller is a dial-in NetWare client that incorporates PPP software and dial-out hardware, but does not have an Ethernet interface.

  This setting causes the MAX to assign the caller an IPX address derived from the value of IPX Pool#. If the client does not supply its own unique node number, the MAX assigns a unique node number to the client as well. The MAX does not send IPX RIP and SAP advertisements across the connection and ignores IPX RIP and SAP advertisements received from the remote end. However, it does respond to IPX RIP and SAP queries received from dial-in clients.

Ascend-IPX-Route (174)

Description: This attribute enables you to configure a static IPX route in a user profile.

Usage: To configure a static IPX route, use this format:

```
Ascend-IPX-Route="<profile_name> <network#> [<node#>]
[<socket#>] [<server_type>] [<hop_count>] [<tick_count>]
[<server_name>]"
```

Limit each pseudo-user profile to about 25 routes—that is, you should specify up to 25 settings for the Ascend-IPX-Route attribute. The MAX fetches information from each pseudo-user profile in order to gather routing information.

Table C-14 describes each Ascend-IPX-Route argument.

Table C-14. Ascend-IPX-Route arguments

Argument	Description
<profile_name>	Specifies the RADIUS user profile used to reach the network. The default value is null.
<network#>	Indicates the unique internal network number assigned to the NetWare server. The default value is 00000000.
<node#>	Specifies the node number of the NetWare server reached through this route. The default value is 000000000001—the typical node number for a NetWare file server.
<socket#>	Indicates the socket number of the NetWare server reached through this route. Typically, NetWare file servers use socket 0451. The default value is 0000. The number you specify must be a well-known socket number. Services that use dynamic socket numbers may use a different socket each time they load. To bring up a connection to a remote service that uses a dynamic socket number, specify a “master” server that uses a well-known socket number.

Table C-14. Ascend-IPX-Route arguments

Argument	Description
<server_type>	Specifies the SAP service type of the NetWare server. NetWare file servers have SAP service type 0004. The default value is 0000.
<hop_count>	Indicates the distance to the destination network in hops. The default value is 1.
<tick_count>	Specifies the distance to the destination network in IBM PC clock ticks (one-eighteenth of a second). This value is for round-trip timer calculation and for determining the nearest server of a given type.The default value is 12.
<server_name>	Indicates the name of an IPX server. The default value is null.

Dependencies: Each static route must appear in a pseudo-user profile. You create a pseudo-user to store information that the MAX can query—in this case, in order to store IPX routing information. You can configure pseudo-users for both global and MAX-specific configuration control of IPX dialout routes. The MAX loads the unit-specific dialout routes in addition to the global dialout routes.

For a unit-specific IPX dialout route, specify the first line of a pseudo-user entry in this format:

```
ipxroute-<unit_name>-<num> Password="ascend", Service-
Type=Outbound
```

For a global IPX dialout route, specify the first line of a pseudo-user entry in this format:

ipxroute-<num> **Password="ascend", Service-Type=Outbound**

<unit_name> is the system name of the MAX—that is, the name specified by the Name parameter in the System Profile. <num> is a number in a sequential series, starting at 1.

In each pseudo-user entry, you can specify one or more routes using the Ascend-IPX-Route attribute. When you have properly configured the profile, RADIUS adds IPX dialout routes to the routing table whenever you power on or reset the MAX, or when you select the Upd Rem Cfg command from the Sys Diag menu. RADIUS adds the routes in this way:

1 RADIUS looks for entries having the format ipxroute-<unit_name>-1, where
<unit_name> is the system name.

2 If at least one entry exists, RADIUS loads all existing entries having the format
ipxroute-<unit_name>-<num> to initialize the IPX routing table.

The variable <num> is a number in a sequential series, starting with 1.

3 The MAX queries ipxroute-<unit_name>-1, then ipxroute-<unit_name>-2, and so on, until it receives an authentication reject from RADIUS.

4 Once the host-specific routes are loaded, RADIUS loads the global configuration entries; these configurations have the form ipxroute-<num>.

5 The MAX queries ipxroute-1, then ipxroute-2, and so on, until it receives an authentication reject from RADIUS.

Example: This example defines a unit-specific IPX route:

```
ipxroute-CA-1 Password="ascend", Service-Type=Outbound
        Ascend-IPX-Route="def 6 7 8 9 10"
```

This example defines a global IPX route:

```
ipxroute-1 Password="ascend", Service-Type=Outbound
        Ascend-IPX-Route="abc 1 2 3 4 5 "
```

Ascend-Link-Compression (233)

Description: This attribute turns data compression on or off for a PPP link.

Usage: You can specify one of these values:

- Link-Comp-None (0) turns off data compression.

  Link-Comp-None in the default.
- Link-Comp-Stac (1) turns on data compression.

  The MAX applies the STACKER LZS compression/decompression algorithm.

Dependencies: Both sides of the link must set either the Ascend-Link-Compression attribute or the Link Comp parameter to turn on data compression.

Ascend-Maximum-Call-Duration (125)

Description: The Ascend-Maximum-Call-Duration attribute specifies the maximum number of minutes an incoming call can remain connected.

Usage: You can specify an integer between 0 and 1440. The MAX checks the connection once per minute, so the actual time the call is connected is slightly longer than the actual time you set.

The default value is 0 (zero). If you accept the default, the MAX does not set a limit on the duration of an incoming call.

Ascend-Maximum-Channels (235)

Description: This attribute specifies the maximum number of channels allowed on an MP+ call.

Usage: Specify an integer between 1 and the maximum number of channels your system supports. The default value is 1.

Dependencies: Keep this additional information in mind:

- This attribute applies only to MP+ calls.
- For optimum MP+ performance, both sides of a connection must set these parameters and attributes to the same values:

- Base Ch Count (in the Connection Profile) or Ascend-Base-Channel-Count (in RADIUS)
- Min Ch Count (in the Answer Profile and the Connection Profile) or Ascend-Minimum-Channels (in RADIUS)
- Max Ch Count (in the Answer Profile and the Connection Profile) or Ascend-Maximum-Channels (in RADIUS)

Ascend-Maximum-Time (194)

Description: This attribute specifies the maximum length of time in seconds that any session is allowed. Once a session reaches the time limit, its connection is taken off-line.

Usage: Specify an integer between 0 and 4,294,967,295. The default value is 0 (zero); when you accept the default, the MAX does not enforce a time limit.

Ascend-Menu-Item (206)

Description: This attribute defines a single menu item for a user profile. You can specify up to 20 Ascend-Menu-Item attributes per profile. The menu items display in the order in which they appear in the RADIUS profile.

Using this attribute, you can configure the terminal server user profile to give the user a custom menu of items from which to choose. The server uses the custom menu to present the user with a subset of terminal server commands. The user does not have access to the regular menu or to the terminal server command line.

Usage: Enter your specifications using this format:

```
Ascend-Menu Item=<command>;<text>;<match>
```

- <command> is the string sent to the terminal server when the user selects the menu item.
- <text> is the text displayed to the user.
- <match> is the pattern the user must type to select the item.
- The first semi-colon (;) that appears acts as the delimiter between <command> and <text>; the second semi-colon that appears acts as the delimiter between <text> and <match>.

By default, the MAX uses the standard terminal server menu.

Example: Suppose you set these attributes:

```
emma Password="m2dan", Service-Type=Login
   Ascend-Menu-Item="show ip stats;Display IP Stats",
   Ascend-Menu-Item="ping 1.2.3.4;Ping server",
   Ascend-Menu-Item="telnet 10.2.4.5;Telnet to Ken’s
machine",
   Ascend-Menu-Item="show arp;Display ARP Table",
   Ascend-Menu-Selector="               Option:",
   ...
```

The terminal server displays this text:

```
1. Display IP Stats     3. Telnet to Ken’s machine
2. Ping server          4. Display ARP Table.
               Option:
```

Now, suppose you also enter specifications for the <match> option, as in this entry:

```
emma Password="m2dan", Service-Type=Login
   Ascend-Menu-Item="show ip stats;ip=Display ip stats;ip",
   Ascend-Menu-Item="ping 1.2.3.4;p=Ping server. Ctrl-C
stops ping;p",
   Ascend-Menu-Item="telnet 10.2.4.5;t=Telnet to Ken's
machine;t",
   Ascend-Menu-Item="show arp;dsp=Display arp table;dsp ",
   Ascend-Menu-Selector="               Option:",
   ...
```

The terminal server displays this text:

```
ip=Display ip stats         p=Ping server. Ctrl-C stops ping
t=Telnet to Ken's machine    dsp=Display arp table
               Option:
```

Note that you cannot combine numeric menu selections with pattern matching. The first Ascend-Menu-Item attribute determines whether the screen displays numbered selections or patterns. This example shows what you should not do:

```
emma Password="m2dan", Service-Type=Login
   Ascend-Menu-Item="show ip stats;ip=Display ip stats",
   Ascend-Menu-Item="ping 1.2.3.4;p=Ping server. Ctrl-C
stops ping;p",
   Ascend-Menu-Item="telnet 10.2.4.5;t=Telnet to Ken's
machine;t",
   Ascend-Menu-Item="show arp;dsp=Display arp table;dsp ",
   Ascend-Menu-Selector="               Option:",
   ...
```

If you mix numbered selections and pattern matching as in this example, the terminal server screen displays the following text:

```
1. ip=Display ip stats               3. t=Telnet to Ken's
machine
2. p=Ping server. Ctrl-C stops ping  4. dsp=Display arp table
               Option:
```

Ascend-Menu-Selector (205)

Description: This attribute specifies a string as a prompt for user input in the terminal server menu interface.

By default, when you create a custom menu with the Ascend-Menu-Item attribute, the terminal server displays this string when prompting the user to make a selection:

```
Enter Selection (1-<num>, q)
```

The <num> argument represents the last number in the list. The terminal server code automatically determines the value of <num> by determining the number of items in the menu. The only valid user input is in the range 1 through <num>, and q to quit.

However, you can specify a different string for prompting the user to make a selection. The Ascend-Menu-Selector attribute enables you to specify a string that the terminal server displays when prompting a user for a menu selection. If you define this attribute, its value overrides the default of Enter Selection (1-<num>, q).

Usage: Specify a text string containing up to 31 characters. The terminal server displays this string when prompting the user for a menu selection.

Example: Suppose you set these attributes:

```
emma Password="m2dan", Service-Type=Login
   Ascend-Menu-Item="show ip stats;Display IP Stats",
   Ascend-Menu-Item="ping 1.2.3.4;Ping server",
   Ascend-Menu-Item="telnet 10.2.4.5; Telnet to Ken’s
machine",
   Ascend-Menu-Item="show arp;Display ARP Table"
   Ascend-Menu-Selector="               Option:"
```

The terminal server displays this text:

```
1. Display IP Stats     3. Telnet to Ken’s machine
2. Ping server          4. Display ARP Table.
               Option:
```

Note that the valid user input in this example is still 1 through 4, or q to quit.

Ascend-Metric (225)

Description: Ascend-Metric enables you to specify the virtual hop count of an IP route.

If there are two routes available to a single destination network, you can ensure that the MAX uses any available nailed-up channel before using a switched channel by setting the Ascend-Metric attribute to a value higher than the metric of any nailed-up route. The higher the value entered, the less likely that the MAX will bring the link or route on-line. The MAX uses the lowest metric.

Usage: You can specify a number between 1 and 15. This value is the virtual hop count. The default value is 7.

Dependencies: Keep this additional information in mind:

- The Ascend-Metric attribute does not apply to bridged connections, such as Combinet links.
- The hop count includes the metric of each switched link in the route.

Example: If a route to a station takes three hops over nailed-up lines, and Ascend-Metric=4 in a user profile that reaches the same station, the MAX does not bring the user profile’s link on-line. However, if the link is already on-line, the MAX does not use the nailed-up lines.

Ascend-Minimum-Channels (173)

Description: Ascend-Minimum-Channels specifies the minimum number of channels an MP+ call maintains.

Dependencies: You can specify a number between 1 and 32. The default value is 1. Keep this additional information in mind:

- This attribute applies only to MP+ calls.
- For optimum MP+ performance, both sides of a connection must set these parameters and attributes to the same values:
  - Base Ch Count (in the Connection Profile) or Ascend-Base-Channel-Count (in RADIUS)
  - Min Ch Count (in the Answer Profile and the Connection Profile) or Ascend-Minimum-Channels (in RADIUS)
  - Max Ch Count (in the Answer Profile and the Connection Profile) or Ascend-Maximum-Channels (in RADIUS)

Ascend-Modem-PortNo (120)

Description: Specifies, for inclusion in an accounting Stop record, the modem used for the call.

Usage: The MAX sends Ascend-Modem-PortNo as part of an accounting Stop record. The attribute does no appear in a user profile. Because the MAX designates a modem by slot card and port, you must consider the value of Ascend-Modem-SlotNo.

See Also: See Also: Ascend-Modem-SlotNo

Ascend-Modem-ShelfNo (122)

Description: Indicates the number of the shelf that terminates the call.

Usage: The Ascend-Modem-ShelfNo attribute appears in Start records, Stop records, and Checkpoint records.

See Also: "Ascend-Modem-PortNo (120)."
"Ascend-Modem-SlotNo (121)."

Ascend-Modem-SlotNo (Attribute 121)

Description: Specifies, for inclusion in an accounting Stop record, the slot containing the modem used for the call.

The MAX sends Ascend-Modem-SlotNo as part of an accounting Stop record. The attribute does not appear in a user profile.

Dependencies: Because the MAX designates a modem by slot card and port, you must consider the value of Ascend-Modem-PortNo

See Also: Ascend-Modem-PortNo

Ascend-MPP-Idle-Percent (254)

Description: This attribute specifies a percentage of bandwidth utilization below which the MAX clears a single-channel MP+ call.

Usage: Specify an integer between 0 and 99. The default value is 0 (zero); this setting causes the MAX to ignore bandwidth utilization when determining whether to clear a call.

Dependencies: Keep this additional information in mind:

- MP+ must be the selected encapsulation method for the profile (Framed-Protocol=MPP).
- If either end of a connection sets the Ascend-MPP-Idle-Percent attribute or Idle Pct parameter to 0 (zero), the MAX ignores bandwidth utilization when determining when to clear a call.
- Bandwidth utilization must fall below this percentage *on both sides of the connection* before the MAX clears the call.

- If the device at the remote end of the link enters an Ascend-MPP-Idle-Percent or Idle Pct setting lower than the value you specify, the MAX does not clear the call until bandwidth utilization falls below the lower percentage.
- If the time set by the Ascend-Idle-Limit expires, the call disconnects whether or not bandwidth utilization falls below the Ascend-MPP-Idle-Percent setting.
- When bandwidth utilization falls below the Ascend-MPP-Idle-Percent setting, the call disconnects regardless of whether the time specified by the Ascend-Idle-Limit attribute has expired.
- Because the Ascend-MPP-Idle-Percent attribute is dependent on traffic levels on both sides of the connection, we recommend that you use the Ascend-Idle-Limit attribute in preference to it.

Ascend-Multicast-Client (152)

Description: This attribute specifies when the user is a multicast client of the MAX.

To communicate with a multicast (MBONE) router, the MAX acts as a multicast client—it receives queries from the router and responds to them using IGMP (Internet Group Management Protocol). The multicast router may reside on its Ethernet interface or across a WAN link.

To communicate with multicast clients, the MAX sends the clients IGMP queries every 60 seconds, receives responses, and forwards multicast traffic. To the clients it looks like a multicast router, although in fact the MAX is forwarding multicast packets based on group memberships.

Usage: You can specify one of these values:

- Multicast-No (0)

  This setting indicates that the user is not a multicast client of the MAX.
- Multicast-Yes (1)

  This setting indicates that the user is a multicast client of the MAX.

Dependencies: Applies solely to the IP-only release of the MAX 4000.

Ascend-Multicast-GLeave-Delay

Description: Specifies the number of seconds the MAX TNT waits before forwarding an IGMP version `2 leave group` message from a multicast client.

Usage: Specify a number of seconds from - to 120. The default is 0 (zero). If you specify a value other than the default, and the MAX TNT receives a `leave group` message, the unit sends an IGMP query to the WAN interface or client from which it receives the `leave group` message. If the MAX TNT does not receive a response from an active multicast client from the same group, it sends a `leave group` message when the time you specify expires.

If you accept the default, the MAXD TNT forwards a `leave group` message immediately. If users might establish multiple multicast sessions for identical groups, set Ascend-Multicast-GLeave-Delay to a value of 10 to 20 seconds.

Dependencies: Ascend-Multicast-GLeave-Delay applies only if you set Multicast-Forwarding=Yes in the IP-Global profile, and Multicast-Allowed=Yes in the IP-Interface profile.

Ascend-Multicast-Rate-Limit (153)

Description: This attribute specifies how many seconds the MAX waits before accepting another packet from a multicast client. To prevent multicast clients from creating response storms to multicast transmissions, you configure the user profile to limit the rate at which the MAX accepts packets from clients.

Usage: Specify an integer. If you set the attribute to 0 (zero), the MAX does not apply rate limiting. The default value is 5. Any subsequent packets received in that 5-second window are discarded.

Dependencies: Applies solely to the IP-only release of the MAX 4000.

Ascend-Multilink-ID (187)

Description: This attribute specifies the ID number of the Multilink bundle when the session closes. A Multilink bundle is a multichannel MP or MP+ call. Each on-line channel of the MP or MP+ call is a session. Ascend-Multilink-ID is sent in an Accounting-Request packet when all of these conditions are true:

- The session was authenticated.

- The session has ended (Acct-Status-Type=Stop).
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Ascend-Multilink-ID does not appear in a user profile and has no default value.

Ascend-Netware-timeout (223)

Description: This attribute specifies how long in minutes the MAX responds to NCP watchdog requests on behalf of IPX clients on the other side of an off-line IPX bridging connection. Responding to watchdog requests on behalf of clients is commonly called watchdog spoofing.

Usage: Specify an integer between 0 and 65535. The default value is 0 (zero). This default allows the MAX to respond to watchdog requests without a time limit.

The timer begins counting down as soon as the WAN bridging link goes off-line. At the end of the selected time, the client-server connections are released. If there is a re-connection of the WAN session, the timeout is cancelled.

Dependencies: Ascend-Netware-timeout applies to IPX bridging connections when the MAX is on the server LAN and not on the client LAN—that is, when Ascend-Handle-IPX=
Handle-IPX-Server.

Ascend-Number-Sessions (202)

Description: This attribute specifies the number of active user sessions of a given class (as specified by the Class attribute). In the case of multichannel calls, such as MP+ calls, each separate connection counts as a session.

Usage: The value of this attribute is the number of sessions that are active for the specified class. The Ascend-Number-Sessions attribute has a compound value. The first part specifies a user-session class; the second part reports the number of active sessions in that class.

In the MAX, you can set the Sess Timer parameter in the Auth submenu of the Ethernet Profile to send accounting requests at regular intervals. At the specified interval, the MAX reports the number of open sessions by sending an Ascend-

Access-Event-Request packet (code 33). This packet contains two attributes—the NAS-Identifier attribute (4), followed by a list of Ascend-Number-Sessions attributes (202).

Dependencies: The Ascend-Number-Sessions attribute is sent in Ascend-Access-Event-Request packets. Only RADIUS daemons customized to recognize this packet code respond these request packets from the MAX. Other accounting daemons ignore it. Therefore, both the standard Livingston RADIUS daemon and the Ascend accounting daemon ignore this attribute.

When modifying the accounting daemon, make sure that it recognizes an Ascend-Access-Event-Request packet in this format:

```
Code (8-bit)=33
Identifier (8-bit) defined in the RADIUS Accounting Draft
Length (16-bit) defined in the RADIUS Accounting Draft
Authenticator (48-bit) defined in the RADIUS Accounting
Draft
List of Ascend-Number-Sessions attributes
```

Example: Suppose that the MAX has three classes of clients: Class-1, Class-2, and Class-3. At the time of the sessions report, there are eight active sessions: three Class-1 sessions, four Class-2 sessions, and one Class-3 session. The accounting packet sent back to the RADIUS accounting server has three Ascend-Number-Session attributes, one for each of these class/session pairs.

Ascend-Numbering-Plan-ID (105)

Specifies the NumberPlanID field in the called party's information element. NumPlanID is used for outbound calls made by the MAX TNT or PRI lines so that the switch can properly interpret the phone number dialed.

Usage: Ask your PRI provider for details on when to use each of the following settings:

- Unknown-Numbering-Plan (0) specifies that NumberPlanID=0.
- ISDN-Numbering-Plan (1, the default) specifies that NumberPlanID=1.
- Private-Numbering-Plan specifies that NumberPlanID=9.

Ascend-Num-In-Multilink (188)

Description: This attribute specifies the number of sessions remaining in a Multilink bundle when the session closes. A Multilink bundle is a multichannel MP or MP+ call. Each on-line channel of the MP or MP+ call is a session.

Ascend-Num-In-Multilink is sent in an Accounting-Request packet when all of these conditions are true:

- The session was authenticated.
- The session has ended (Acct-Status-Type=Stop).
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Ascend-Num-In-Multilink does not appear in a user profile and has no default value.

Ascend-PPP-Address (253)

Description: This attribute specifies the MAX unit's IP address reported to the calling unit during PPP IPCP negotiations.

Usage: Specify an IP address in dotted decimal notation. An IP address consists of four numbers between 0 and 255, separated by periods. The default value is 0.0.0.0. If you accept the default, IPCP negotiates with the value of the IP Adrs parameter in the Ethernet Profile.

If you specify a valid IP address, IPCP negotiates with that IP address. If you specify 255.255.255.255, IPCP negotiates with the address 0.0.0.0.

Dependencies: You can assign Ascend-PPP-Address a value different from the MAX unit's true IP address, as long as the user requesting access understands that limitation.

Ascend-PPP-Async-Map (212)

Description: This attribute gives the Ascend PPP code the async control character map for the PPP session. The control characters are passed through the PPP link as data and are used only by applications running over the link.

Usage: Specify a 4-byte bitmap to one or more control characters. The async control character map is defined in RFC 1548 and specifies that each bit position represents its ASCII equivalent. The bits are ordered with the lowest bit of the lowest byte being 0. For example, bit 19 corresponds to Control-S (DC3) or ASCII 19.

Example: Your specification might look like this one:

```
emma Password="m2dan", Service-Type=Login
     Ascend-PPP-Async-Map=19,
     ...
```

The number 19 translates to 13 hex or 10011 binary. Therefore, NUL (00), SOH (01), and EOT (04) are mapped.

Ascend-PPP-VJ-1172 (211)

Description: This attribute instructs the Ascend PPP code to use the 0x0037 value for the VJ compression type. The MAX uses this value only during IPNCP negotiation. Incoming 1172 type options are accepted without this option being set.

RFC 1172 section 5.2 contains an erroneous statement that the VJ compression type value is 0x0037; it should be 0x002d. However, many older PPP implementations use the 0x0037 value when negotiating VJ compression. If you do not specify a value for Ascend-PPP-VJ-1172, the VJ compression type is 0x002d.

Usage: Enter your specification using this format:

```
Ascend-PPP-VJ-1172=PPP-VJ-1172
```

Ascend-PPP-VJ-Slot-Comp (210)

Description: This attribute instructs the Ascend PPP code not to use slot compression when sending VJ-compressed packets.

When you turn on VJ compression, the MAX removes the TCP/IP header, and associates a TCP/IP packet with a connection by giving it a slot ID. The first packet coming into a connection must have a slot ID, but succeeding packets need not have one. If the packet does not have a slot ID, the MAX assumes that it should be associated with the last-used slot ID. This scenario uses slot ID compression, because the slot ID is not used in any packet but the first in a stream.

However, there may be times when you want each VJ-compressed packet to have a slot ID. The Ascend-PPP-VJ-Slot-Comp attribute exists for this purpose.

Usage: To specify that no slot compression occurs, set the Ascend-PPP-VJ-Slot-Comp attribute to VJ-Slot-Comp-No (1). If you do not specify a value for Ascend-PPP-VJ-Slot-Comp, and Framed-Compression=Van-Jacobson-TCP-IP, slot compression occurs.

Ascend-Preempt-Limit (245)

Description: This attribute specifies the number of idle seconds the MAX waits before using one of the channels of an idle link for a new call.

Usage: Specify an integer between 0 and 65535. The MAX never preempts a call if you enter 0 (zero). The default value is 60.

Dependencies: The Ascend-Preempt-Limit attribute does not apply to nailed-up links.

Ascend-Pre-Input-Octets (190)

Description: This attribute indicates the number of input octets before authentication.

Ascend-Pre-Input-Octets is included in an Accounting-Request packet when all of these conditions are true:

- The session was authenticated.

- The session has ended (Acct-Status-Type=Stop).
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Ascend-Pre-Input-Octets does not appear in a user profile. Its default value is 0 (zero).

Ascend-Pre-Input-packets (192)

Description: This attribute indicates the number of input packets before authentication.

Ascend-Pre-Input-packets is included in an Accounting-Request packet when all of these conditions are true:

- The session was authenticated.
- The session has ended (Acct-Status-Type=Stop).
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Ascend-Pre-Input-packets does not appear in a user profile. Its default value is 0 (zero).

Ascend-Pre-Output-Octets (191)

Description: This attribute indicates the number of output octets before authentication.

Ascend-Pre-Output-Octets is included in an Accounting-Request packet when all of these conditions are true:

- The session was authenticated.
- The session has ended (Acct-Status-Type=Stop).
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Ascend-Pre-Output-Octets does not appear in a user profile. Its default value is 0 (zero).

Ascend-Pre-Output-packets (193)

Description: This attribute indicates the number of output packets before authentication.

Ascend-Pre-Output-packets is included in an Accounting-Request packet when all of these conditions are true:

- The session was authenticated.
- The session has ended (Acct-Status-Type=Stop).
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Ascend-Pre-Output-packets does not appear in a user profile. Its default value is 0 (zero).

Ascend-PreSession-Time (198)

Description: This attribute reports the length of time in seconds from when a call connected to when it completes authentication.

Ascend-PreSession-Time is included in an Accounting-Request packet when both of these conditions are true:

- The session has ended or has failed to authenticate (Acct-Status-Type=Stop).
- The Auth parameter is not set to RADIUS/LOGOUT.

Usage: Ascend-PreSession-Time does not appear in a user profile. Its default value is 0 (zero).

Ascend-PRI-Number-Type (226)

Description: This attribute specifies the type of phone number the MAX dials.

Usage: You can specify one of these values:

- Unknown-Number (0)

  This setting indicates that the MAX can dial any type of number. Intl-Number (1)

  This setting indicates that the MAX dials a number outside the U.S.
- National-Number (2)

This setting indicates that the MAX dials a number inside the U.S. National-Number is the default.

- Local-Number (4)

  This setting indicates that the MAX dials a number within your Centrex group.

- Abbrev-Number (5)

  This setting indicates that the MAX dials an abbreviated phone number.

Ascend-Primary-Home-Agent (129)

Description: The Ascend-Primary-Home-Agent attribute specifies the first home agent the foreign agent tries to reach when setting up an ATMP tunnel, and indicates the UDP port the foreign agent uses for the link.

The RADIUS server passes the attributes in the mobile node's RADIUS user profile to the foreign agent. The foreign agent sends these attributes when connecting with the home agent.

A mobile node can also connect directly to the home agent. An ATMP-based RADIUS profile that is local to the home agent enables the mobile node to bypass a foreign agent connection, but does not preclude a foreign agent. If both the home agent and the foreign agent have local RADIUS profiles for the mobile node, the node can choose between a direct connection or a tunneled connection through the foreign agent.

Usage: Specify the primary home agent using this syntax:

```
Ascend-Primary-Home-Agent=" hostname | ip_address [:
udp_port]"
```

- The hostname argument indicates the home agent's symbolic hostname.
- The ip_address argument indicates the home agent's IP address in dotted decimal notation.

  Specify an IP address if a DNS server is not set up for the home agent. You can specify a hostname or an IP address, but not both.
- The optional udp_port argument indicates the UDP port on which the foreign agent communicates with the home agent.
- The default value is 5150.

- The colon (:) separates the hostname or IP address from the UDP port specification.

Example: To specify the home agent max1.home.com at IP address 10.0.0.1, and indicate that the foreign agent should use UDP port 6001, specify one of these lines in the RADIUS user profile:

```
Ascend-Primary-Home-Agent="max1.home.com:6001"
```

```
Ascend-Primary-Home-Agent="10.0.0.1:6001"
```

The following RADIUS profile authenticates a mobile NetWare client that connects directly to the home agent. In this example, the home agent is in gateway mode. It forwards packets from the mobile node across a nailed-up WAN link to the home IPX network.

```
Mobile-IPX Password="unit"
   User-Service=Framed,
   Ascend-Route-IPX=Route-IPX-Yes,
   Framed-Protocol=PPP,
   Ascend-IPX-Peer-Mode=IPX-Peer-Dialin,
   Framed-IPX-Network=40000000,
   Ascend-IPX-Node-Addr=12345678,
   Ascend-Primary-Home-Agent="max1.home.com:6001",
   Ascend-Secondary-Home-Agent="max2.home.com:6001",
   Ascend-Home-Network-Name="Dave’s MAX",
   Ascend-Home-Agent-Password="Pipeline"
```

Dependencies: Keep this additional information in mind:

- If you specify the Ascend-Home-Agent-UDP-Port attribute on the line immediately following the Ascend-Primary-Home-Agent attribute, you need not specify a value for ***udp_port.***

  By the same token, if you specify a value for the udp_port argument of Ascend-Secondary- Home-Agent, or if you accept the default of 5150, you need not specify the Ascend-Home-Agent-UDP-Port attribute.
- It is preferable to use Ascend-Primary-Home-Agent in place of the Ascend-Home-Agent-IP- Addr attribute in the RADIUS user profile. However, the Stop record will include Ascend-Home-Agent-IP-Addr and not Ascend-Primary-Home-Agent.

- To specify a secondary home agent for use if the primary home agent is unavailable, use the Ascend-Secondary-Home-Agent attribute.

Note: The RADIUS accounting Stop record will include Ascend-Home-Agent-IP-Addr when Ascend-Primary-Home-Agent is present in the user profile.

See Also: "Ascend-Home-Agent-Password (184)."
"Ascend-Home-Agent-UDP-Port (186)."
"Ascend-Home-Network-Name (185)."
"Ascend-Secondary-Home-Agent (130)."

Ascend-Private-Route (104)

Description: Specifies a destination address and next-hop router address for a private route. A RADIUS user profile can specify a list of private routes associated with the connection. The private routes affect only packets received from the connection. (The routes are not added to the global routing table.) If a destination is not found in the list of private routes and there is no default private route, the global routing table is consulted for a decision on routing the packets. Otherwise, only the private routing table is consulted.

Usage: In a user profile, specify the attribute in the following format: Ascend-Private-Route=" dest_addr/netmask next_hop/netmask" where dest_addr/netmask is the destination address of the route, and next_hop/netmask is the address of the next-hop router.

Example: Following is a sample user profile that creates three private routes associated with the caller:

```
pipe50 Password="ascend"
User-Service=Framed,
Framed-Protocol=PPP,
Framed-IP-Address=10.1.1.1,
Framed-IP-Netmask=255.0.0.0,
Ascend-Private-Route="170.1.0.0/16 10.10.10.1"
Ascend-Private-Route="200.1.1.1/32 10.10.10.2"
scend-Private-Route="20.1.0.0/16 10.10.10.3"
Ascend-Private-Route="0.0.0.0/0 10.10.10.4"
```

With this profile, the private routing table for the connection contains the following routes, including a default route: Dest/Mask Gateway

170.1.0.0/16 10.10.10.1
200.1.1.1/32 10.10.10.2
20.1.0.0/16 10.10.10.3
0.0.0.0/0 10.10.10.4

Dependencies: The user profile can also specify the Ascend-Client-Gateway attribute, but the specification will not modify the private default route if one has been specified via the Ascend-Private-Route attribute.

See Also: “Ascend-Client-Gateway (132).”

Expiration (1032)

Description: The Expiration attribute specifies an expiration date for a user’s password in a user profile

When the MAX makes an authentication request, the RADIUS server checks the current date against the value of Expiration. If the date of the authentication request is the same date or a later date than the value of Expiration, the user receives a message saying that the password has expired.

You must specify Expiration when you first create a user.

Usage: Specify a month, day, and year.

- For the month specification, enter the first three letters of the month in which you want the password to expire, or specify the entire name of the month The month must begin with a capital letter. For the day specification, enter one or more digits indicating a valid day of the month The values 2, 02, 002, and 0021 are all valid, but 32 is not.
- For the year specification, enter a four-digit year.

  The year must start with the number 19.
- Separate each part of the date specification using one or more spaces, tabs, or commas.

The default value is 00/00/00.

Dependencies: Keep this additional information in mind:

- If a password expires and the user resets it, the RADIUS server adds the value of Ascend-PW-Lifetime to the date on which the user resets the password. The resulting date becomes the new value for Expiration.

  For example, suppose that Ascend-PW-Lifetime=30, Expiration=January 1, 1997, and today's date is March 1, 1997. If the user resets the password today, the value of Expiration becomes today's date + Ascend-PW-Lifetime, or March 31, 1997.
- If the password has not expired, the value of Expiration overrides the value of Ascend-PW-Lifetime.

For example, if on January 1, 1997 you set Ascend-PW-Lifetime=30 and Expiration=January 15, 1997, the password expires on January 15, 1997. In other words, if the password has not expired, the value of Ascend-PW-Lifetime is irrelevant.

Example: Your specification might look like this one:

```
Emma Password="m2dan", User-Service=Login, Expiration=" January 1, 1997"
...
```

See Also: See Also: "Ascend-PW-Lifetime (208)"

Ascend-PW-Expiry

Note: Ascend-PW-Expiry is no longer supported. The attribute that replaced Ascend-PW-Expiry is Expiration.

Ascend-PW-Lifetime (208)

Description: This attribute specifies the number of days that a password is valid.

Usage: Specify an integer to indicate the number of days for which the user's password is valid. You can set the Ascend-PW-Lifetime attribute on any line other than the first line of the user profile.

Dependencies: Keep this additional information in mind:

- If a password expires and the user resets it, the RADIUS server adds the value of Ascend-PW-Lifetime to the date on which the user resets the password; the resulting date becomes the new value for Expiration.

  For example, suppose that Ascend-PW-Lifetime=30, Expiration=January 1, 1996, and today's date is March 1, 1996. If the user resets the password today, the value of Expiration becomes today's date + Ascend-PW-Lifetime, or March 31, 1996.
- If the password has not expired, the value of Expiration overrides the value of Ascend-PW-Lifetime.

  For example, if on January 1, 1996 you set Ascend-PW-Lifetime=30 and Expiration=January 15, 1996, the password expires on January 15, 1996. In other words, if the password has not expired, the value of Ascend-PW-Lifetime is irrelevant.
- If Ascend-PW-Lifetime is absent, the value of Password-Expiration determines the password duration.

  The Password Expiration value in the RADIUS dictionary is the default value for Ascend-PW-Lifetime. By default, the Password-Expiration value is 30.

Example: You might make this specification:

```
emma Password="m2dan", Service-Type=Login, Expiration="Jan
1, 1996"
    Ascend-PW-Lifetime=30
```

Ascend-PW-Warntime (207)

Description: Specifies the number of days before password expiration that the RADIUS server sends a message informing the user that the password will expire. The message appears when the user establishes a connection, and is carried to the MAX TNT in the Reply-Message attribute.

Usage: Specify an integer. If Ascend-PW-Warntime is absent, the value of Password-Warning determines the warntime. The Password-Warning value in the RADIUS dictionary is the default value for Ascend-PW-Warntime. By default, the Password-Warning value is 5.

Example: Suppose you set Ascend -PW-Warntime=5. Starting five days before the expiration of the password, the RADIUS server sends a message telling the user the number of days until the password expires.

Dependencies: Note that the user might never see a warning message, even though the RADIUS server returns the message to the MAX TNT. This situation can occur if the user is using PPP for authentication (rather than the terminal server), or using a script to exchange information with the terminal server.

See Also: “Expiration (1032)”
“Ascend-PW-Lifetime (208)

Ascend-Receive-Secret (215)

Description: This attribute specifies a value that must match the password that the RADIUS server sends it to your MAX from the calling unit.

Usage: You can use the Ascend-Receive-Secret attribute for CACHE-TOKEN or PAP-TOKEN-CHAP authentication. In either case, you can specify up to 20 characters. The default value is null.

- CACHE-TOKEN authentication uses a shared secret, and simplifies the authentication process by caching the user’s token for the fixed length of time specified by the Ascend-Token-Expiry attribute; during the lifetime of the token, subsequent calls by the user require only CHAP authentication without the use of a hand-held security card.

  For this type of authentication, set the Ascend-Receive-Secret attribute to the same password as the Send PW parameter in the Connection Profile that places the incoming call. The RADIUS server uses this value to authenticate incoming calls from a user while his or her token is cached and alive. The cached token is deposited on the MAX during the initial security-card authentication process.

- PAP-TOKEN-CHAP authentication uses an encrypted CHAP password with which the answering unit authenticates second and subsequent channels of an MP+ call.

  For type of authentication, Set Ascend-Receive-Secret to the value of the Aux Send PW parameter in the Connection Profile used to dial the call.

  In PAP-TOKEN-CHAP authentication, you need to verify only the initial connection using a hand-held security card. CHAP verifies any additional channels. That is, whenever the MAX adds channels to a PPP or MP+ call using PAP-TOKEN-CHAP, the calling unit sends the encrypted value of Aux Send PW, and the answering unit checks this password against Ascend-

Receive-Secret. The answering unit receives Ascend-Receive-Secret from the RADIUS server when the first channel of the call connects.

Example: This example shows the settings necessary for a user called “John” to use an ACE server. The password received from the user is sent to the security server for authentication.

```
John  Authentication-Type=ACE, Ascend-Token-Expiry=90,
Ascend-Token-Idle=80, Ascend-Token-Immediate=Tok-Imm-Yes
      Ascend-Receive-Secret="shared-secret",
      Service-Type=Framed,
      Framed-Protocol=MPP,
      Framed-IP-Address=200.0.5.1,
      Framed-IP-Netmask=255.255.255.0
```

This example shows the settings necessary for a user called “Emma” to use an ACE server. Because this entry includes the attribute Ascend-Receive-Secret, the MAX can authenticate additional channels through CHAP without having to go to the ACE server for authentication.

```
Emma  Authentication-Type=ACE
      Service-Type=Framed,
      Framed-Protocol=MPP,
      Framed-IP-Address=200.0.5.1,
      Framed-IP-Netmask=255.255.255.0,
      Ascend-Receive-Secret="b5XSAM"
```

Ascend-Redirect-Number (93)

Description: Indicates the redirected number extracted from the Redirect Number Information Element (IE) in an ISDN frame. If the IE is present, this number is sent to the RADIUS server for each Start and Stop accounting request. If the IE is not present in the frame, the attribute is not sent to the RADIUS server

Usage: You can use the Redirect Number Information Element in an ISDN frame to bill dial-in clients according to the original called number. This Information Element is generated by a Public Switched Telephone Network (PSTN) switch when the phone number dialed by a customer has been redirected to an another number.

Ascend-Remote-Addr (155)

Description: This attribute specifies the IP address of the numbered interface at the remote end of a link.

Usage: Specify the IP address of the numbered interface. An IP address consists of four numbers between 0 and 255, separated by periods. The default value is 0.0.0.0.

Dependencies: For Ascend-Remote-Addr to apply, you must enable IP for the user profile (Ascend-Route-IP=Route-IP-Yes).

Ascend-Remove-Seconds (241)

Description: This attribute specifies the number of seconds that average line utilization (ALU) for transmitted data must fall below the threshold indicated by the Ascend-Target-Util attribute before the MAX begins removing bandwidth from a session. The MAX determines the ALU for a session by using the algorithm specified by the Ascend-History-Weigh-Type attribute.

When utilization falls below the threshold for a period of time greater than the value of the Ascend-Remove-Seconds attribute, the MAX attempts to remove the number of channels specified by the Ascend-Dec-Channel-Count attribute. Using the Ascend-Add-Seconds and Ascend-Remove-Seconds attributes prevents the system from continually adding and subtracting bandwidth, and can slow down the process of allocating or removing bandwidth.

Usage: Specify a number between 1 and 300. The default value is 10.

Dependencies: Keep this additional information in mind:

- One channel must be up at all times.
- Removing bandwidth cannot cause the ALU to exceed the threshold specified by the Ascend-Target-Util attribute.
- The number of channels remaining cannot fall below the amount specified by the Ascend-Minimum-Channels attribute.
- Ascend-Add-Seconds and Ascend-Remove-Seconds have little or no effect on a system with a high Ascend-Seconds-Of-History value.

  If the value of Ascend-Seconds-Of-History is low, the Ascend-Add-Seconds and Ascend-Remove-Seconds attributes provide an alternative way to ensure that spikes must persist for a certain period of time before the system responds.

Ascend-Require-Auth (201)

Description: This attribute specifies whether additional authentication is required after CLID (Calling Line ID) authentication.

Usage: You can specify one of these values:

- Not-Require-Auth (0) specifies that additional authentication is not required. Not-Require-Auth is the default.

  Require-Auth (1) specifies that additional authentication is required.

  If you require additional authentication, you must configure a two-tiered dial-in setup. The first-tier dial-in user profile has the following two-line format:

```
<phonenum> Password="Ascend-CLID" Service-Type=Outbound
            Ascend-Require-Auth=Require-Auth
```

  - <phonenum> represents the calling party's phone number.
  - The Password setting specifies that RADIUS authenticates the caller by caller ID.
  - The Service-Type setting indicates that the entry does not allow dial-in users.
  - The Ascend-Require-Auth setting specifies that after CLID authentication, additional authentication is required.

  When you set Ascend-Require-Auth=Require-Auth, you should not include any other attributes in the user profile. You must specify the characteristics of the call in the second-tier user profile.

Example: This example shows a two-tiered approach. The first user profile specifies CLID authentication, and indicates that additional authentication will follow. Because Recv Auth=CHAP in the Answer Profile, CHAP authentication will follow CLID authentication. The second user profile sets up other attributes for the call.

```
5551212   Password="Ascend-CLID" Service-Type=Outbound
          Ascend-Require-Auth=Require-Auth
```

```
Emma   Password="pwd" Calling-Station-ID="5551212"
       Service-Type=Framed,
       Framed-Protocol=PPP,
       Framed-IP-Address=200.11.12.10,
       Framed-IP-Netmask=255.255.255.248,
       Ascend-Send-Secret="pwd",
       ...
```

Ascend-Route-Appletalk (118)

Description: Specifies whether AppleTalk routing is enabled for the connection. When AppleTalk routing is enabled, the connection can forward AppleTalk packets.

Usage: Specify one of the following values:

- Route-Appletalk-No (0) disables AppleTalk routing for this user profile.
- Route-Appletalk-Yes (1) enables AppleTalk routing for this user profile.
  The default is No (0).

Dependencies: : If you specify Route-Appletalk-Yes, you must set the Ascend-Appletalk-Peer-Mode attribute.

See Also: Ascend-Appletalk-Peer-Mode (117)
Ascend-Appletalk-Route(116)

Ascend-Route-IP (228)

Description: This attribute specifies whether IP routing is allowed for the user profile.

Usage: You can specify one of these values:

- Route-IP-No (0)
- Route-IP-Yes (1)
  Route-IP-Yes is the default.

Ascend-Route-IPX (229)

Description: This attribute indicates whether IPX routing is allowed for the user profile. For PPP and MP+ calls, both ends of the connection must have matching settings to route IPX.

Usage: You can specify one of these values:

- Route-IPX-No (0)

  Route-IPX-No is the default.
- Route-IPX-Yes (1)

Ascend-Route-Preference (126)

Description: Specifies the preference for a route defined by the Framed-IP-Address attribute in a user profile. Every RADIUS user profile that specifies an explicit IP address using the Framed-IP-Address attribute indicates a static route.

Usage: Specify an integer. The default value is 60. Ascend recommends that you accept the default.

Ascend-Secondary-Home-Agent (130)

Description: The Ascend-Secondary-Home-Agent attribute specifies the secondary home agent the foreign agent tries to reach when the primary home agent (specified by Ascend-Primary- Home-Agent) is unavailable. The attribute also indicates the UDP port the foreign agent uses for the link.

Usage: Specify the secondary home agent using this syntax:

```
Ascend-Secondary-Home-Agent=" hostname | ip_address [:
udp_port]"
```

- The hostname argument indicates the home agent's symbolic hostname.
- The ip_address argument indicates the home agent's IP address in dotted decimal notation.

  Specify an IP address if a DNS server is not set up for the home agent. You can specify a hostname or an IP address, but not both.
- The optional udp_port argument indicates the UDP port on which the foreign agent communicates with the home agent.

The default value is 5150.

- The colon (:) separates the hostname or IP address from the UDP port specification.

Example: To specify max2.home.com at IP address 10.0.0.2 as the secondary home agent, and indicate that the foreign agent should use UDP port 6002, specify one of these lines in the RADIUS user profile:

```
ascend-Weconday-Home-Agent=”max2.home.com:6002”
```

```
Ascend-Secondary-Home-Agent="10.0.0.2:6002"
```

To specify a primary home agent and a secondary home agent, enter these lines in the RADIUS user profile:

```
Ascend-Primary-Home-Agent="max1.home.com:6001"
```

```
Ascend-Secondary-Home-Agent="max2.home.com:6002"
```

The foreign agent first tries max1.home.com on UDP port 6001. If the name cannot be resolved, or if max1.home.com does not respond, the foreign agent then tries max2.home.com on UDP port 6002.

The RADIUS accounting Stop record will include Ascend-Home-Agent-IP-Addr when Ascend-Secondary-Home-Agent is present in the user profile.

Dependencies: If you specify the Ascend-Home-Agent-UDP-Port attribute on the line immediately following the Ascend-Secondary-Home-Agent attribute, you need not specify a value for udp_port. By the same token, if you specify a value for the udp_port argument of Ascend-Secondary-Home-Agent, or if you accept the default of 5150, you need not specify the Ascend-Home-Agent-UDP-Port attribute.

See Also: “Ascend-Home-Agent-Password (184).”
“Ascend-Home-Agent-UDP-Port (186).”
“Ascend-Home-Network-Name (185).”
“Ascend-Primary-Home-Agent (129).”

Ascend-Seconds-Of-History (238)

Description: This attribute specifies the number of seconds the MAX uses as a sample for calculating average line utilization (ALU) of transmitted data; the MAX arrives at this average using the algorithm specified by the Ascend-History-Weigh-Type attribute.

The number of seconds you choose for the Ascend-Seconds-Of-History attribute depends on your device's traffic patterns. For example, if you want to average spikes with normal traffic flow, you may want the MAX to establish a longer historical time period. If, on the other hand, traffic patterns consist of many spikes that are short in duration, you may want to specify a shorter period of time; doing so assigns less weight to the short spikes.

Usage: Specify a number between 1 and 300. The default value is 15 seconds.

Dependencies: Keep this additional information in mind:

- Ascend-Seconds-Of-History applies only to MP+ calls (Framed-Protocol=MPP).
- If you specify a small value for the Ascend-Seconds-Of-History attribute, and increase the values of the Ascend-Add-Seconds attribute and the Ascend-Remove-Seconds attribute relative to the value of Ascend-Seconds-Of-History, the system becomes less responsive to quick spikes.

  The easiest way to determine the proper values for all these attributes is to observe usage patterns; if the system is not responsive enough, the value of Ascend-Seconds-Of-History is too high.

Ascend-Send-Auth (231)

Description: This attribute specifies the authentication protocol that the MAX requests when initiating a connection using PPP or MP+ encapsulation. The answering side of the connection determines which authentication protocol, if any, the connection uses.

Usage: You can specify one of these values:

- Send-Auth-None (0) indicates that the MAX does not request an authentication protocol for outgoing calls.

  Send-Auth-None is the default.

- Send-Auth-PAP (1) indicates that the MAX requests PAP (Password Authentication Protocol).

  PAP is a PPP authentication protocol that provides a simple method for the MAX to establish its identity in a two-way handshake. Authentication takes place only upon initial link establishment, and does not use encryption. The remote device must support PAP.

  If you specify this setting, the MAX requests PAP authentication, but uses CHAP authentication if the called unit requires CHAP. Choose this setting for non-token card authentication if you want to send your password unencrypted.

- Send-Auth-CHAP (2) indicates that the MAX requests CHAP (Challenge Handshake Authentication Protocol).

  CHAP is a PPP authentication protocol that is more secure than PAP. CHAP provides a way for the remote device to periodically verify the identity of the MAX using a three-way handshake and encryption. Authentication takes place upon initial link establishment; a device can repeat the authentication process any time after the connection is made. The remote device must support CHAP.

  If you specify this setting, the MAX does not bring up the connection using PAP. Choose this setting for non-token card authentication if you do not wish to send your password unencrypted—that is, if you do not wish to use PAP authentication.

Dependencies: Keep this additional information in mind:

- Ascend-Send-Auth is applicable only to outgoing user profiles in RADIUS.
- The link must use PPP or MP+ encapsulation.
- If you request PAP or CHAP authentication, you must also specify a password using Ascend-Send-Secret or Ascend-Send-Passwd.

Ascend-Send-Passwd (232)

Description: This attribute specifies the password that the RADIUS server sends to the remote end of a connection on an outgoing call.

Usage: Specify a text string containing up to 20 characters. The default value is null.

Dependencies: In a user profile, you can specify either Ascend-Send-Passwd or Ascend-Send-Secret, but not both. Use Ascend-Send-Passwd only if your version of the MAX does not support Ascend-Send-Secret.

Ascend-Send-Secret (214)

Description: This attribute specifies the password that the RADIUS server sends to the remote end of a connection on an outgoing call. It is encrypted when passed between the RADIUS server and the MAX.

Usage: Specify a text string containing up to 20 characters. The default value is null.

Dependencies: In a user profile, you can specify either Ascend-Send-Passwd or Ascend-Send-Secret, but not both. Use Ascend-Send-Passwd only if your version of the MAX does not support Ascend-Send-Secret.

Ascend-Session-Svr-Key (151)

Description: The Ascend-Session-Svr-Key attribute enables the MAX to match a user session with a client request to perform certain operations, such as disconnecting a session or changing a session's filters.

The client sends Ascend-Session-Svr-Key to the RADIUS server in a Disconnect-Request or Change-Filter-Request packet when it initiates an operation. In addition, Ascend-Session-Svr-Key appears in a RADIUS Accounting-Start packet when a session starts.

Usage: Specify up to 16 characters. The default value is null.

Dependencies: The client sends the Ascend-Session-Svr-Key attribute only if Session Key=Yes in the Ethernet>Mod Config>RADIUS Server menu.

Ascend-Shared-Profile-Enable (128)

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

Ascend-Source-Auth (103)

Description: Specifies a source IP address and associated billing code. RADIUS can look up a billing code on the basis of the source IP address of a packet. When the MAX TNT places a call on behalf of a packet with the specified source address, it also sends the associated billing code to the network switch. This feature is referred to as Source Auth.

Because looking up an IP address resembles a route lookup, this feature uses some of the same mechanisms as static routes. For example, Source Auth entries are retrieved from RADIUS when the router is initialized and the Source Auth information is cached for later use. The Source Auth entries can be refreshed by using the new Refresh –s command

Usage: In a user profile or pseudo-user profile, make your specification in the following format:

```
Ascend-Source-Auth=" address/mask – authcode"
```

where address/mask is the source address and subnet mask, and authcode is the billing code conveyed to the switch when a call is placed on behalf of a packet from the given source address.

As with static routes, you can indicate the subnet mask with any desired level of specificity, and the most specific entry prevails in case of conflict. The maximum length of an authcode is the same as the maximum for Ascend-Billing-Number: 24 digits. The hyphen (–) delimiter is reserved for future capabilities.

Example: Ascend-Source-Auth="10.150.0.0/16 - 5105551212"

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.

Ascend-Source-IP-Check (96)

Description: Enables or disables anti-spoofing for this session.

Usage: Specify one of the following settings:

- Source-IP-Check-No (0) disables anti-spoofing. This setting is the default.
- Source-IP-Check-Yes (1) specifies that the system checks all packets received on this interface to ensure that the source IP address in the packets matches the far-end remote address or the address agreed upon in IPCP negotiation. If the addresses do not match, the system discards the packet.

Example: In the following RADIUS user profile, anti-spoofing is enabled:

```
ed-mc1-p75 Password="localpw", User-Service=Framed
Framed-Protocol=PPP,
Framed-IP-Address=10.7.8.200,
Framed-IP-Netmask=255.255.255.0,
Ascend-Source-IP-Check=Source-IP-Check-Yes
```

Ascend-Target-Util (234)

Description: This attribute specifies the percentage of bandwidth use at which the MAX adds or subtracts bandwidth.

Usage: Specify an integer between 0 and 100. The default value is 70.When the value is 70%, the device adds bandwidth when it exceeds a 70 percent utilization rate, and subtracts bandwidth when it falls below that number.

Dependencies: Keep this additional information in mind:

- When selecting a target utilization value, keep these guidelines in mind:
  - Monitor how the application behaves when using different bandwidths.

    For example, an application might be able to use 88% of a 64-kbps link, but only 70% of a 256-kbps link.
  - Monitor the application at different loads.
- Ascend-Target-Util applies only if the link is using MP+ encapsulation (Framed-Protocol=MPP).

Ascend-Telnet-Profile (91)

Description: Specifies the name of the Security profile to use for the authenticated Telnet session.

Usage: Specify the name of a Security profile. The default is null.

Example: `Ascend-Telnet-Profile=”Full Access”`

See Also: Ascend-Host-Info (252), Login-IP-Host (14)

Ascend-Third-Prompt (213)

Description: In the MAX configuration interface, the 3rd Prompt parameter enables you to specify an additional prompt for user input after the login and password prompts in the terminal server interface. The MAX passes the information the user enters to the RADIUS server as the Ascend-Third-Prompt attribute.

Usage: The Ascend-Third-Prompt attribute can contain up to 80 characters and does not appear in a user profile. If the user enters more than 80 characters, the MAX truncates the input to 80. If the user does enter any characters, the MAX sets the attribute to null.

Ascend-Token-Expiry (204)

Description: This attribute specifies the lifetime in minutes of a cached token.

CACHE-TOKEN authentication uses a shared secret, and simplifies the authentication process by caching the user’s token for the fixed length of time specified by the Ascend-Token-Expiry attribute.When the cached token is still alive, CHAP authenticates subsequent CACHE-TOKEN access requests from the same user without the use of a hand-held security card. When the cached token has expired, the ACE server authenticates CACHE-TOKEN access requests.

Usage: On the first line of the user profile, specify an integer representing the lifetime of the cached token in minutes. The default value is 0 (zero). If you accept the default, the MAX rejects subsequent CACHE-TOKEN requests from the same user.

Example: The following two-line example allows CACHE-TOKEN authentication with a 90-minute token cache. Notice that the Ascend-Token-Expiry attribute must appear on the first line of the entry, along with the username and Authentication-Type=ACE or = Defender:

```
Connor  Authentication-Type =ACE, Ascend-Token-Expiry=90
       Password="ACE",
       Ascend-Receive-Secret="shared-secret",
        ...
```

Ascend-Token-Idle (199)

Description: This attribute specifies the maximum length of time in minutes a cached token can remain alive between authentications.

Usage: On the first line of the user profile, specify an integer representing the maximum length of time in minutes that a cached token can remain alive. The default value is o (zero). If you accept this default, the cached token remains alive until the value of the Ascend-Token-Expiry attribute causes it to expire.

Dependencies: Typically, the value of Ascend-Token-Idle is lower than the value of Ascend-Token-Expiry.

Example: The following two-line example allows CACHE-TOKEN authentication with a 90-minute token cache and an 80-minute idle limit. Notice that the Ascend-Token-Idle attribute must appear on the first line of the entry:

```
Jim  Authentication-Type=ACE, Ascend-Token-Expiry=90,
Ascend-Token-Idle=80
       Password=bowie
```

```
        Ascend-Receive-Secret="shared secret"
```

Ascend-Token-Immediate (200)

Description: This attribute specifies how RADIUS treats the password received from a login user when the user profile specifies a hand-held security card server. Use this attribute in an ACE user profile that contains the setting Service-Type=Login.

Usage: You can specify one of these values:

- Tok-Imm-No (0) indicates that the password received from the user is ignored.

  Choose this value for a security server that requires that a user enter a challenge using a security card before the security server derives a password.

  Tok-Imm-No is the default.
- Tok-Imm-Yes (1) specifies that the password received from the user is sent to the security server for authentication.

Dependencies: The Ascend-Token-Immediate attribute does not work with CHAP authentication.

Example: This example shows a portion of a user profile that sends the password received from the login user to the ACE server. The login derives the password from a hand-held security card:

```
Connor  Authentication-Type=ACE, Ascend-Token-Immediate=Tok-
Imm-Yes
        Password=terminate
Ascend-Receive-Secret="shared-secret",
        Service-Type=Login,
          ...
```

Ascend-Transit-Number (251)

Description: This attribute specifies the U.S Interexchange Carrier (IEC) you use for long distance calls over a T1 PRI line.

Usage: Specify the same digits you use to prefix a phone number dialed over an ISDN BRI line, T1 access line, or voice interface:

- 288 selects AT&T.
- 222 selects MCI.
- 333 selects Sprint.

The default value is null. If you accept the default, the MAX uses any available IEC for long-distance calls.

Ascend-TS-Idle-Limit (169)

Description: This attribute specifies the number of seconds that a terminal server connection must be idle before the MAX disconnects the session.

Usage: You can specify a value between 0 and 65535.The default is 120. A setting of 0 (zero) means that the line can be idle indefinitely.

Dependencies: Ascend-TS-Idle-Limit does not apply if you are using a Frame Relay or raw TCP connection, or if Ascend-TS-Idle-Mode=TS-Idle-None.

Ascend-TS-Idle-Mode (170)

Description: This attribute specifies whether the MAX uses a terminal server idle timer and, if so, whether both the user and host must be idle before the MAX disconnects the session.

Usage: You can specify one of these settings:

- TS-Idle-None (0)

  This setting indicates that the MAX does not disconnect the session no matter how long the line is idle. This setting disables the idle timer.
- TS-Idle-Input (1)

This setting indicates that the MAX disconnects the session if the user is idle for a length of time greater than the value of the Ascend-TS-Idle-Limit attribute.

TS-Idle-Input is the default.

- TS-Idle-Input-Output (2)

  This setting indicates that the MAX disconnects the session if both the user and the host are idle for a length of time greater than the value of the Ascend-TS-Idle-Limit attribute.

Example: This entry specifies that the user must be idle for 90 seconds before the MAX disconnects the session.

```
DEFAULT Password="UNIX"
       Service-Type=Login,
       Ascend-TS-Idle-Limit=90,
       Ascend-TS-Idle-Mode=TS-Idle-Input
```

Dependencies: Ascend-TS-Idle-Mode does not apply if you are using a Frame Relay or raw TCP connection.

Ascend-User-Acct-Base (142)

Description: The Ascend-User-Acct-Base attribute specifies whether the numeric base of the RADIUS Acct-Session-ID attribute is 10 or 16.

Usage: Specify one of these settings:

- Ascend-User-Acct-Base-10 indicates that the numeric base is 10.

  The default is 10.
- Ascend-User-Acct-Base-16 indicates that the numeric base is 16.

For example, when you set Ascend-User-Acct-Base=Ascend-User-Acct-Base-10, the MAX presents a typical session ID to the accounting server in this way:

"1234567890"

When you set Ascend-User-Acct-Base=Ascend-User-Acct-Base-16, the MAX presents the same session ID in this way:

"499602D2"

Dependencies: Changing the value of Ascend-User-Acct-Base while sessions are active results in inconsistent reporting between the Start and Stop records.

See Also: “Ascend-User-Acct-Host (139).”
“Ascend-User-Acct-Key (141).”
“Ascend-User-Acct-Port (140).”
“Ascend-User-Acct-Time (143).”
“Ascend-User-Acct-Type (138).”

Ascend-User-Acct-Host (139)

Description: The Ascend-User-Acct-Host attribute specifies the IP address of the RADIUS accounting server to use for this connection.

Usage: Specify an IP address in dotted decimal notation n.n.n.n, where n is an integer between 0 and 255. The default value is 0.0.0.0.

See Also: “Ascend-User-Acct-Base (142)”
“Ascend-User-Acct-Key (141)”
“Ascend-User-Acct-Port (140)”
“Ascend-User-Acct-Time (143)
“Ascend-User-Acct-Type (138).”

Ascend-User-Acct-Key (141)

Description: The Ascend-User-Acct-Key attribute specifies the RADIUS client password as it appears in the clients file.

Usage: Specify a text string. The default value is null.

See Also: “Ascend-User-Acct-Base (142).”
“Ascend-User-Acct-Host (139).”
“Ascend-User-Acct-Port (140).”
“Ascend-User-Acct-Time (143).”
“Ascend-User-Acct-Type (138).”

Ascend-User-Acct-Port (140)

Description: The Ascend-User-Acct-Port attribute specifies a UDP port number for the connection between the user and the RADIUS accounting server.

Usage: Specify the UDP port number you indicated for the authentication process of the daemon in /etc/services. Or, if you used the incr keyword to the –A option when starting the daemon, specify the number of the UDP port for authentication services +1. You can specify a number between 1 and 32767.

See Also: “Ascend-User-Acct-Base (142).”
“Ascend-User-Acct-Host (139).”
“Ascend-User-Acct-Key (141).”
“Ascend-User-Acct-Time (143).
“Ascend-User-Acct-Type (138).”

Ascend-User-Acct-Time (143)

Description: The Ascend-User-Acct-Time attribute specifies the number of seconds the MAX waits for a response to a RADIUS accounting request from the RADIUS accounting server for this connection.

Usage: Specify an integer between 1 and 10. The default value is 0 (zero).

See Also: “Ascend-User-Acct-Base (142).”
“Ascend-User-Acct-Host (139).”
“Ascend-User-Acct-Key (141).”
“Ascend-User-Acct-Port (140).”
“Ascend-User-Acct-Type (138).”

Ascend-User-Acct-Type (138)

Description: Specifies the RADIUS accounting server(s) to use for the connection.

Usage: Specify one of the following settings:

- Ascend-User-Acct-None (0) specifies the MAX TNT sends accounting information to the RADIUS server specified by the Acct-Server parameter. This server is known as the default server. Ascend-User-Acct-None is the default.

- Ascend-User-Acct-User (1) specifies that the MAX TNT sends accounting information to the RADIUS server specified by the Ascend-User-Acct-Host attribute in the RADIUS user profile.
- Ascend-User-Acct-User-Default (2) specifies that the MAX TNT sends accounting information both to the RADIUS server specified by the Ascend-User-Acct-Host attribute in the RADIUS user profile, and to the default server.

See Also: "Ascend-User-Acct-Base (142)."
"Ascend-User-Acct-Host (139)."
"Ascend-User-Acct-Key (141)."
"Ascend-User-Acct-Port (140)."
"Ascend-User-Acct-Time (143)."

Ascend-VRouter-Name (102)

Description: Specifies the name of a defined Virtual Router (VRouter). Specifying the VRouter name in a RADIUS user profile groups the WAN interfaces with the VRouter.

Usage: Specify the name of a VRouter. The default is null, which specifies that the global VRouter is in use.

Example: The following user profiles specifies a VRouter called Corpa:

```
bob Password="bob"
User-Service=Framed,
Framed-Protocol=PPP,
Ascend-VRouter-Name="Corpa"
```

See Also: "Ascend-IP-Pool-Definition (217)."
"Framed-Route (22)."

Ascend-Xmit-Rate (255)

Description: Specifies the transmit baud rate for the connection.

Dependencies: : The Ascend-Xmit-Rate attribute is sent in Accounting-Request packets at the end of a session under these conditions:

- he Accounting-Request packet has Acct-Status-Type=Stop.
- The Auth parameter is set to a value other than RADIUS/LOGOUT.

The attribute is still sent with the Accounting-Request packet whether the connection is authenticated or not.

Authentication-Type (1027)

Description: This attribute specifies the method used to authenticate a user.

Usage: The value of the Authentication-Type attribute determines the method NavisRadius uses to verify the identity of a user requesting network access. You can specify one of these methods of authentication in a user profile: None, Passwd, RADIUS, TACACS, Realm, Local, File, TACPLUS, ARA-DES, ACE, DEFENDER, SKEY, WINNT, and DNIS-REALM.

RADIUS example

```
dawson Authentication-Type=WinNT,
Service-Type=Framed,
Framed-IP-Address=137.157.8.8
```

Dependencies: This attribute must be a check-item. Check-items must appear on the first line of a user profile.

Called-Station-ID (30)

Description: Called-Station-ID specifies the called-party number, indicating the phone number dialed by the user to connect to the MAX. Called-Station-ID is set only if the called-party number is known.

Usage: Called-Station-ID is set by the MAX and sent in Access-Request, Access-Accept, and Accounting-Request packets. This attribute does not appear in a user profile and has no default value.

Calling-Station-ID (31)

Description: This attribute specifies the calling party number for CLID (Calling Line ID) authentication, indicating the phone number of the user that wants to connect to the MAX.

- If you set CLID Auth=Prefer in the Answer Profile, the MAX checks the calling party's phone number against the value of the Calling-Station-ID attribute whenever CLID authentication is available.

  If a match is found, and no further authentication is required, the MAX accepts the call.
- If you set CLID Auth=Required in the Answer Profile, the calling party's phone number must match the value of the Calling-Station-ID attribute before the MAX can answer the call.

  If CLID is not available, the MAX does not answer the call.

Usage: Specify a telephone number. You can indicate up to 37 characters, limited to the following:

```
1234567890()[]!z-*#|
```

The default value is null.

Example: This user profile is configured for CLID authentication using name, password, and caller ID:

```
Emma  Password="test", Calling-Station-ID="123456789"
      Service-Type=Framed,
      Framed-Protocol=PPP,
      Framed-IP-Address=255.255.255.254,
      Framed-IP-Netmask=255.255.255.255,
      Ascend-Assign-IP-Pool=1,
      Ascend-Route-IP=Route-IP-Yes,
      Ascend-Idle-Limit=30
```

CHAP-Password (3)

Description: This attribute specifies the response value provided by a CHAP (Challenge Handshake Authentication Protocol) user in response to the password challenge.

Usage: CHAP-Password is set by the MAX and sent in Access-Request packets. The default value is null.

Class (25)

Description: This attribute enables access providers to classify user sessions, such as for the purpose of billing users depending on the service option they choose.

If you include the Class attribute in the RADIUS user profile, the RADIUS server sends it to the MAX in the Access-Accept packet when the session begins. Class is then included in Accounting-Request packets sent to the RADIUS accounting server under these conditions:

- Whenever a session starts
- Whenever a session stops (as long as the Auth parameter is not set to RADIUS/LOGOUT)

Keep in mind that the accounting entries give the class on a per-user and per-session basis. The Ascend-Number-Sessions attribute reports information on all user sessions—that is, on the number of current sessions of each class.

In addition, suppose the MAX starts CLID authentication by sending an Access-Request packet and receives the Class attribute in an Access-Accept packet. If the MAX requires further authentication, it includes Class in the Access-Request packet.

Usage: Specify an alphanumeric text string containing up to 253 characters. The default value is null.

Expiration (21)

Description: This attribute specifies an expiration date for a user's password in a user profile.

When the MAX makes an authentication request, the RADIUS server checks the current date against the value of Expiration. If the date of the authentication request is the same date or a later date than the value of Expiration, the user receives a message saying that the password has expired.

You must specify Expiration when you first create a user.

Usage: Specify a month, day, and year.

- For the month specification, enter the first three letters of the month in which you want the password to expire; or, you can specify the entire name of the month.

  The month must begin with a capital letter.
- For the day specification, enter one or more digits indicating a valid day of the month; 2, 02, 002, and 0021 are all valid, but 32 is not.
- For the year specification, enter a four-digit year.

  The year must start with the number 19.
- Separate each part of the date specification using one or more spaces, tabs, or commas.

The default value is 00/00/00.

Dependencies: Keep this additional information in mind:

- If a password expires and the user resets it, the RADIUS server adds the value of

Ascend-PW-Lifetime to the date on which the user resets the password; the resulting date becomes the new value for Expiration.

For example, suppose that Ascend-PW-Lifetime=30, Expiration=January 1, 1996, and today's date is March 1, 1996. If the user resets the password today, the value of Expiration becomes today's date + Ascend-PW-Lifetime, or March 31, 1996.

- If the password has not expired, the value of Expiration overrides the value of Ascend-PW-Lifetime.

  For example, if on January 1, 1996 you set Ascend-PW-Lifetime=30 and Expiration=January 15, 1996, the password expires on January 15, 1996. In other words, if the password has not expired, the value of Ascend-PW-Lifetime is irrelevant.

Example: Your specification might look like this one:

```
emma Password="m2dan"
        Service-Type=Login, Expiration="November 1, 1995"
```

Framed-IP-Address (8)

Description: This attribute specifies the IP address of the caller in a user profile.

RADIUS can authenticate an incoming call by matching its IP address to one specified in the RADIUS user profile. In addition, if the remote end requires an IP address on an outgoing call, and does not assign one dynamically, you must specify it in the user profile.

Usage: Specify an IP address in dotted decimal notation *n.n.n.n*, where *n* is an integer between 0 and 255. The default value is 0.0.0.0. An answering user profile with this setting matches all IP addresses.

Dependencies: Every Connection Profile and RADIUS user profile that specifies an explicit IP address is a static route.

Framed-Compression (13)

Description: This attribute turns TCP/IP header compression on or off.

Usage: To turn on TCP/IP header compression, specify Van-Jacobson-TCP-IP. This setting applies only to packets in TCP applications, such as Telnet, and turns on header compression for both sides of the link. By default, this attribute does not turn on header compression.

Dependencies: Turning on header compression is most effective in reducing overhead when the data portion of the packet is small.

Framed-IPX-Network (23)

Description: This attribute specifies a virtual IPX network required for the ATMP (Ascend Tunnel Management Protocol) home agent to route IPX packets to the mobile node. When specified in a user profile, the Framed-IPX-Network attribute instructs the answering unit to advertise an additional IPX route.

Usage: Specify the IPX network number of the IPX router at the remote end of the connection. The default value is null.

RADIUS requires that Framed-IPX-Network have a decimal value (base 10), but IPX network numbers generally appear as hexadecimal values (base 16). In order to give this attribute a value, you must convert the hexadecimal IPX network number to decimal format for use in the user profile. For example, if the IPX network number is 13870000, you must convert it to the decimal 49990000. This requirement does not apply for the IPX node address, which is represented as a 12-digit string enclosed in double-quotes.

Framed-MTU (12)

Description: This attribute specifies the maximum number of bytes the MAX can receive in a single packet on a PPP, Frame Relay, EU-UI, or EU-RAW link.

Usage: The default value is 1524; you should accept this default unless the device at the remote end of the link cannot support it. If the administrator of the remote network specifies that you must change this value, specify a number between 1 and 1524 (for a PPP, EU-UI, or EU-RAW link) or between 128 and 1600 (for a Frame Relay link).

Framed-IP-Netmask (9)

Description: This attribute specifies a subnet mask for the caller at Framed-IP-Address in a user profile.

Usage: Specify an IP address in dotted decimal notation *n.n.n.n,* where *n* is an integer between 0 and 255. The default value is 0.0.0.0. If you accept this default, the MAX assumes a default netmask based on the "class" of the address as shown in ,Table C-15.

Table C-15. IP address classes and default netmasks

Class	**Address range**	**Network bits**
Class A	0.0.0.0 → 127.255.255.255	8
Class B	128.0.0.0 → 191.255.255.255	16
Class C	192.0.0.0 → 223.255.255.255	24
Class D	224.0.0.0 → 239.255.255.255	N/A
Class E (reserved)	240.0.0.0 → 247.255.255.255	N/A

Framed-Protocol (7)

Description: This attribute specifies the type of framed protocol the link can use. When you set this attribute, the link cannot use any other type of framed protocol.

Usage: Table C-16 lists the values you can specify for Framed-Protocol.

Table C-16. Framed-Protocol settings

Setting	Incoming call	Outgoing call
PPP (1)	A user requesting access can dial in using MP+ (Multilink Protocol Plus), MP (Multilink Protocol), or PPP (Point-to-Point Protocol) framing. A user requesting access can also dial in unframed, and then change to PPP framing. If the user dials in using any other type of framing, the MAX rejects the call.	Outgoing calls use PPP framing.
SLIP (2)	A user requesting access can dial in unframed and change to SLIP framing. SLIP requires that a user dial in without using a framed protocol before changing to SLIP.	This value does not apply to outgoing calls.
MPP (256)	This value does not apply to incoming calls.	Outgoing calls request MP+ framing.
EURAW (257)	A user requesting access can dial in using EURAW framing. EURAW is a type of X.75 encapsulation in which IP packets are HDLC encapsulated with a CRC field. If the user dials in using any other type of framing, the MAX rejects the call.	Outgoing calls use EURAW framing.

Table C-16. Framed-Protocol settings

Setting	Incoming call	Outgoing call
EUUI (258)	A user requesting access can dial in using EUUI framing. EUUI is a type of X.75 encapsulation in which IP packets are HDLC encapsulated with a CRC field and a small header. If the user dials in using any other type of framing, the MAX rejects the call.	Outgoing calls use EUUI framing.
COMB (260)	A user requesting access can dial in using Combinet framing. If the user dials in using any other type of framing, the MAX rejects the call.	Outgoing calls use Combinet framing.
FR (261)	This value does not apply to incoming calls.	Outgoing calls use frame relay (RFC 1490) framing.
Ascend-ARA (262)	A dial-in user can establish an ARA (AppleTalk Remote Access) connection to the Ethernet network.	This value does not apply to outgoing calls.
FR-CIR (263)	This value specifies a frame relay circuit.	This value specifies a frame relay circuit.

Note: By default, the MAX does not limit the protocols a link can access.

Dependencies: What Framed-Protocol does depends on how you set Service-Type:

- If Service-Type=Framed or is unspecified, a user requesting access can dial in using the framing specified by Framed-Protocol; the MAX rejects other types of framing.

  A user requesting access can also dial in without using a framed protocol, but can then change to the framing specified by the Framed-Protocol attribute.

- If Service-Type=Framed or is unspecified, and Framed-Protocol has no specified value, the operator can use any framed protocol.
- If Service-Type=Login, the user cannot use a framed protocol.
- If Service-Type=Outbound, Framed-Protocol specifies the type of framing allowed on the outgoing call.

Example: The dial-in user in this example cannot use the terminal server and is limited to PPP protocols (PPP, MP+, or MP).

```
ascend Password="pipeline"
       Service-Type=Framed,
       Framed-Protocol=PPP,
       Framed-IP-Address=10.0.200.225,
       Framed-IP-Netmask=255.255.255.0,
       Ascend-Metric=2,
       Framed-Routing=None,
       Framed-Route="10.0.220.0 10.0.200.225 1",
       Ascend-Idle-Limit=30
```

The dial-in user in this example establishes an ARA connection to the Ethernet network:

```
ascend Password="pipeline"
       Service-Type=Framed,
       Framed-Protocol=Ascend-ARA
       Ascend-Idle-Limit=30,
...
```

Framed-Route (22)

Description: This attribute enables you to add static IP routes to the MAX unit's routing table.

Usage: The Framed-Route attribute has this format:

```
Framed-Route="<host_ipaddr>[/<subnet mask>] <gateway_ipaddr>
<metric> [<private>] [<name>]"
```

You should limit each pseudo-user profile to about 25 routes—that is, you should specify up to 25 settings for the Framed-Route attribute in a pseudo-user profile. The MAX fetches information from each entry in order to initialize its routing table.

Table C-17 describes each Framed-Route argument.

Table C-17. Framed-Route arguments

Syntax element	Description
<host_ipaddr>/<subnet_mask>	Indicates the IP address of the destination host or subnet reached by this route. If the address includes a subnet mask, the remote router specified by <router_ipaddr> is a router to that subnet, rather than to a whole remote network. To specify the entire remote network, do not specify a subnet mask.
<router_ipaddr>	Specifies the IP address of the router at the remote end of the connection. The 0.0.0.0 address is a wildcard entry replaced by the caller's IP address.When RADIUS authenticates a caller and sends the MAX an Access-Accept message with a Framed-Route 0.0.0.0 router, the MAX updates its routing tables with the Framed-Route value, but substitutes the caller's IP address for the router. This setting is especially useful when RADIUS cannot know the IP address of the caller because the IP address is assigned from an address pool.
<metric>	Indicates the metric for this route. If the MAX has more than one possible route to a destination network, it chooses the one with the lower metric.
<private>	Specifies "y" if this route is private, or "n" if it is not private. If you specify that the route is private, the MAX does not disclose the existence of the route when queried by RIP or another routing protocol.
<name>	Indicates the name outgoing user profile that uses the route.

Dependencies: Each static route must appear in a pseudo-user profile. You create a pseudo-user to store information that the MAX can query—in this case, in order to store IP routing information. You can configure pseudo-users for both global and MAX-specific configuration control of IP dialout routes. The MAX loads the unit-specific dialout routes in addition to the global dialout routes.

For a unit-specific IP dialout route, specify the first line of a pseudo-user entry in this format:

```
route-<unit_name>-<num> Password="ascend", Service-Type=Outbound
```

For a global IP dialout route, specify the first line of a pseudo-user entry in this format:

```
route-<num> Password="ascend", Service-Type=Outbound
```

<unit_name> is the system name of the MAX—that is, the name specified by the Name parameter in the System Profile. <num> is a number in a sequential series, starting at 1.

In each pseudo-user entry, you can specify one or more routes using the Framed-Route attribute. When you have properly configured the profile, RADIUS adds IP dialout routes to the routing table whenever you power on or reset the MAX, or when you select the Upd Rem Cfg command from the Sys Diag menu. RADIUS adds the routes in this way:

1. RADIUS looks for entries having the format route-<unit_name>-1, where <unit_name> is the system name.
2. If at least one entry exists, RADIUS loads all existing entries having the format
   route-<unit_name>-<num> to initialize the IP routing table.

   The variable <num> is a number in a sequential series, starting with 1.
3. The MAX queries route-<unit_name>-1, then route-<unit_name>-2, and so on, until it receives an authentication reject from RADIUS.
4. Once the host-specific routes are loaded, RADIUS loads the global configuration entries; these configurations have the form route-<num>.

5 The MAX queries route-1, then route-2, and so on, until it receives an authentication reject from RADIUS.

The routes remain in effect until the next restart or until overwritten by dynamic updates or routes specified in Connection Profiles.

Note: In some cases, you might wish to update the MAX unit's routing tables when connecting to a user whose profile specified Service-Type=Framed. In this case, you can set the Framed-Route attribute in an incoming user profile to specify the user's IP address and subnet mask in the <host_ipaddr> and <subnet_mask> arguments; the route you specify in this manner exists only during the time the call is on-line. When you enter a nonzero router address for <router_ipaddr> that is different from the caller's address, the static route of a dial-in framed persists even after the connection goes off-line.

Example: This example shows two RADIUS pseudo-user profiles defining global static IP routes:

```
route-1   Password="ascend" Service-Type=Outbound
   Framed-Route="10.0.200.33/29 10.0.200.37 1 n lala-gw-out
"
   Framed-Route="10.0.200.50/29 10.0.200.37 1 n lala-gw-out
"
   Framed-Route="10.0.200.47/29 10.0.200.49 1 n nana-gw-out
"

route-2   Password="ascend" Service-Type=Outbound
   Framed-Route="11.0.200.33/29 11.0.200.37 1 n zzz-gw-out "
   Framed-Route="12.0.200.47/29 11.0.200.49 1 n kk-gw-out "
```

Framed-Routing (10)

Description: This attribute specifies whether the MAX sends RIP (Routing Information Protocol) packets, receives RIP packets, or both.

Usage: You can specify one of these values:

- None (0) indicates that the MAX does not send or receive RIP updates.

  None is the default. Many sites turn off RIP on the WAN interface in order to avoid storing very large local routing tables. If you turn off RIP, the MAX does not listen to RIP updates across the connection. To route to other networks through that connection, the MAX must rely on static routes specified in a pseudo-user profile.

- Broadcast (1) indicates that the MAX sends RIP version 1 updates, but does not receive them.
- Listen (2) indicates that the MAX receives RIP version 1 updates, but does not send them.
- Broadcast-Listen (3) indicates that the MAX both sends and receives RIP version 1 updates.
- Broadcast-v2 (4) indicates that the MAX sends RIP version 2 updates, but does not receive them.
- Listen-v2 (5) indicates that the MAX receives RIP version 2 updates, but does not send them.
- Broadcast-Listen-v2 (6) indicates that the MAX both sends and receives RIP version 2 updates.

Dependencies: If RIP is enabled to both send and receive RIP updates on the WAN interface, the MAX broadcasts its routing table to the remote network and listens for RIP updates from that network. Gradually, all routers on both networks have consistent routing tables (all of which may become quite large).

Group-Limit-Name (1040)

Description: Specifies the name of the group to which a user belongs. The Group-Limit-Name may be applied to the DEFAULT user.

Usage: Enter the attribute/value pair in the first line of the user's profile in the users file.

Example*:*

```
user1 Password="test", Group-Limit-Name="group1", Group-Ses-
sion-Limit=100
Service-Type = Framed,
Framed-Protocol = PPP,
Framed-IP-Address = 192.0.2.1,
Framed-IP-Netmask = 255.255.255.0
DEFAULT Authentication-Type=UNIX, Group-Limit-Name="group2",
Group-Session-Limit= 25
Service-Type = Framed,
Framed-Protocol = PPP,
Ascend-Route-IP = Route-IP-Yes,
Ascend-Metric = 2,
Framed-Routing = None,
Ascend-Assign-IP-Pool = 2
```

Dependencies: The Group-Limit-Name must be included as a check-item in the user profile of any user that belongs to a group.

See Also: Group-Session-Limit, Group-Name

Group-Session-Limit (104)

Description*:* Specifies the total number of active sessions a group can have across the network served by a RADSTATE daemon. Not a required check-item in the user profiles of all members of the group. More connections by a group's users than are specified by the attribute's value are permitted if the user profiles of individual members of the group contain the Group-Limit-Name attribute but do not contain the Group-Session-Limit attribute. May be entered in a user profile that contains the Simultaneous-Use attribute that limits the number of simultaneous sessions an individual can have on the network.

Usage*:* Enter the attribute/value pair in the first line of a group member's user's profile in the users file if you want the user's request for access to be constrained by the group's total of allowable connections.

Example*:*

```
user238 Password = "test", Simultaneous-Use = 2, Group-
Limit-Name = "group3", Group-Session-Limit = 200
Service-Type = Framed,
Framed-Protocol = PPP,
Ascend-Route-IP = Route-IP-Yes,
Ascend-Metric = 2,
Framed-Routing = None,
Ascend-Assign-IP-Pool = 3
```

Dependencies: Must be used with the Group-Limit-Name attribute.

- Can use with REALMs or DNIS-REALMS if the authfile authentication type field is changed to FILE and you create a prefix.users file that contains a DEFAULT user profile that has the Group-Limit-Name, Group-Session-Limit attributes.

See Also: Group-Limit-Name, Simultaneous-Use

Login-IP-Host (14)

Description: This attribute specifies the IP host to which the user automatically connects when you set Service-Type=Login and specify a value for the Login-Service attribute. Access begins immediately after login.

Usage: Specify an IP address in dotted decimal notation *n.n.n.n*, where *n* is an integer between 0 and 255. The default value is 0.0. 0.0.This setting specifies that the Login does not automatically connect to a particular host.

If you do not specify a value for the Login-IP-Host attribute, the user can access any remote host through the Telnet or raw TCP commands of the terminal server command-line interface. When the operator uses the menu-driven terminal server interface, access to remote hosts is limited to the hosts listed by the Ascend-Host-Info attribute.

Dependencies: Keep this additional information in mind:

- Login-IP-Host has the same functionality as the <hostname> field in the terminal server command-line interface.

  Closing the remote terminal server session also automatically closes the session with the Login-IP-Host.
- When Service-Type=Framed, RADIUS ignores the Login-IP-Host attribute.

Login-Service (15)

Description: This attribute specifies the type of terminal service connection to an IP host that occurs immediately after authentication.

Usage: Specify one of these values:

- Telnet (0)

The user immediately establishes a Telnet session with the host specified by the Login-IP-Host attribute.

- Rlogin (1)

  The user immediately establishes an Rlogin session with the host specified by the Login-IP-Host attribute.

- TCP-Clear (2)

  This setting specifies a TCP/IP connection with no Telnet protocol. TCP-Clear establishes a TCP session between the MAX and the host specified by Login-IP-Host over which the user can run an application specified by Login-TCP-Port.

  If you specify this setting, the Answer Profile must specify TCP-Clear=Yes.

When you set the Login-Service attribute, a dial-in terminal server user makes an immediate connection to an IP host on your local network and never sees the terminal server interface.

By default, the MAX does not grant immediate access to an IP host.

Dependencies: Keep this additional information in mind:

- If you specify both Login-Service and Login-IP-Host, the MAX automatically connects the Login to the host specified by Login-IP-Host.
- If you do not specify Login-Service or Login-IP-Host, the Login sees either the MAX unit’s terminal server command-line interface or the terminal server menu interface, depending upon how the MAX is configured.

Example: In this example, an Rlogin session starts automatically for anyone using the “userx” username and “xyzzy” password. When the session terminates, the connection also terminates.

```
# This profile causes an auto-rlogin to 10.0.200.4 upon
login.
userx  Password="xyzzy"
       Service-Type=Login,
       Login-Service=Rlogin,
       Login-IP-Host=10.0.200.4
```

Further, when you specify the following settings, a raw TCP session starts automatically for anyone using the “user1” username and “test1” password:

```
# This profile causes an auto-TCP to 4.2.3.1 port 9 upon
login.
user1  Password="test1"
```

```
        Service-Type=Login,
        Login-Service=TCP-Clear,
        Login-IP-Host=4.2.3.1,
        Login-TCP-Port=9
```

Login-TCP-Port (16)

Description: This attribute specifies the port number to which a TCP session connects when Login-Service=TCP-Clear in a user profile.

Usage: Specify an integer between 1 and 65535. The default value is 23.

Dependencies: Login-TCP-Port has the same functionality as the <port-number> field in the MAX unit's terminal server command-line interface. For information on the terminal server command-line interface, see the *MAX ISP and Telecommuting Configuration Guide*.

NAS-IP-Address (4)

Description: This attribute indicates the IP address of the MAX. When the MAX sends an Access-Request packet, it indicates its IP address to the RADIUS server using this attribute.

Usage: In most cases, you never need to specify the NAS-IP-Address attribute in a user profile.

However, you might want to specify it if multiple MAX units use a single RADIUS server, and you want to specify the MAX to which a particular user can connect. In this case, the NAS-Identifier value in the Access-Request packet and the NAS-IP-Address value in the user profile must match for the RADIUS server to authenticate the connection.

Specify an IP address in dotted decimal notation *n.n.n.n/nn*, where *n* is an integer between 0 and 255. Suppose that the user "Emma" is allowed to dial into the MAX at IP address 200.65.212.46. The first line of the user profile might look like this one:

```
Emma Password="pwd", NAS-IP-Address=200.65.212.46
```

NAS-Port (5)

Description: This attribute specifies the port on the MAX handling the user session. Specifically, NAS-Port identifies the interface and service the session is using. The MAX sends this attribute to the RADIUS server in an Access-Request packet and an Accounting Request packet.

Usage: You can set the NAS-Port attribute to restrict the line and channel a user can access. On the first line of the user profile, specify NAX-Port using this format:

```
<type> <line> <channel>
```

- <type> can have the value 1 for a digital call, or 2 for an analog call.
- <line> uses two digits to specify the line number the call is using.
- <channel> uses two digits to represent the channel on the line the call is using.

The incoming authentication request must the NAS-Port setting. The default value is 0 (zero).

Example: To restrict a dial-in user to analog service on line 1, channel 0, set up a user profile like this one:

```
name  Password="password", NAS-Port=20100
      User-Name="robin",
      Service-Type=Framed,
      Framed-Protocol=PPP,
      Ascend-Assign-IP-Pool=1,
      Ascend-Route-IP=1,
      Ascend-Idle-Limit=300,
      Framed-Routing=None
```

Old-Password (17)

Description: The MAX and the RADIUS server use this attribute to change an expired password.

When a user specifies an expired password, RADIUS prompts the user for a new password. When the user enters the new password, the MAX sends an Access-Password-Request packet that contains both the old password (as the value of the

Change-Password attribute), and the new password (as the value of the Password attribute).

If the RADIUS server accepts the new password, it tries to edit the users file and replace the expired password with the new one. Note that the RADIUS server can make this change in the user profile only in the flat file. It cannot make this change in the database version of the users file.

Usage: Old-Password does not appear in a user profile and has no default value.

Password (2)

Description: This attribute specifies the password of the calling device or dial-in user in a user profile.

Usage: Specify an alphanumeric string containing up to 252 characters. The default value is null. The Password attribute may be used as a check-item, appearing on the first line, or as a reply-item in the following lines.

For example, consider this first line in a user profile:

```
Emma Password="pwd"
```

The user called “Emma” must specify the password “pwd” in order to gain access to the MAX.

Prohibit (1028)

Description: Prohibit is an authentication check-item. The Prohibit check-item specifies a type of connection which NavisRadius will not authenticate. Several of the values which you can assign the Prohibit attribute perform administrative functions, such as Administrative-User and Authenticate-Only.

Usage: All check-items must appear in the first line of a user profile. You can enter more than one Prohibit attribute in a user profile, provided each is in the first line. You can use the values Prohibit-Authenticate-Only and Prohibit-Administrative-User to perform administrative functions. You can use the value Prohibit-Authenticate-Only to confirm that a user profile will successfully authenticate a user, since no authorization reply-items are returned in the Access-Accept packet. You can also use the value Prohibit-Administrative-User to confirm that a server is on-line. Prohibit-Administrative-User translates to a server-status code and does not prohibit a connection.

RADIUS example

This profile prevents any authenticated connection for a user who is in arrears on service payments.

```
deadbeat Password="randomstring", Prohibit=Prohibit-all
Service-Type=PPP
Framed-IP-Address=137.157.8.8,
...
```

This user profile prevents the authentication of outbound connections for localuser:

```
localuser Password="randomstring", Prohibit-Outbound-User
Service-Type=Outbound-User
Framed-IP-Address=137.157.8.8,
```

Reply-Message (18)

Description: This attribute carries message text from the RADIUS server to RADIUS clients such as the MAX.

- In a pseudo-user profile that configures message text and a list of IP hosts, the Reply-Message attribute specifies text that appears to the terminal server operator who is using the menu-driven interface.
- If the RADIUS server determines that the MAX should terminate the session, it sends an Access-Terminate-Session packet containing the Reply-Message attribute.

Usage: Specify a text string containing up to 80 characters. The default value is null. You can specify up to 16 Reply-Message attributes in a pseudo-user profile.

Dependencies: Keep this additional information in mind:

- An Access-Terminate-Session packet is a RADIUS packet identified by the code number 31; only RADIUS daemons customized to support this packet code can send an Access-Terminate-Session packet.

  Neither the Ascend RADIUS daemon nor the Livingston RADIUS daemon supports this packet type. This packet can include only one attribute—the

Reply-Message attribute—and this attribute can specify up to 80 characters of text.

When the MAX receives an Access-Terminate-Session packet, it starts a timer, displays any Reply-Message included in the packet, and terminates the session. For example, if a user's bill is past due, the Access-Terminate-Session packet could include the message “Emma, you have not paid your connect charges.”

- If you do not specify a Reply-Message attribute in a user profile that authenticates callers, and the RADIUS server sends an Access-Accept packet, no message appears.
- If the RADIUS server sends an Access-Reject packet and you do not specify a Reply-Message attribute in a customized RADIUS daemon, this message appears:

  ```
  ** Bad Password
  ```

  The MAX then allows the user two additional attempts to enter the correct password; if the user does not supply the correct password in three attempts, the MAX terminates the session.
- If the RADIUS server sends an Access-Terminate-Session packet and you do not specify a Reply-Message attribute in a customized RADIUS daemon, the MAX displays this message to the terminal server user:

  ```
  ** Session Terminated
  ```

  The MAX then uses a timer to terminate the login session. The RADIUS server discards all input it received before it terminated the session.

Example: Here is an example of a pseudo-user profile setting up message text for a MAX named Cal:

```
initial-banner-Cal Password="ascend", Service-Type=Outbound
     Reply-Message="Up to 16 lines of up to 80 characters 
each",
     Reply-Message="will be accepted. Long lines will be 
truncated",
     Reply-Message="Additional lines will be ignored.",
     Reply-Message="",
     Ascend-Host-Info="1.2.3.4 Berkeley",
     Ascend-Host-Info="1.2.3.5 Alameda",
     Ascend-Host-Info="1.2.36 San Francisco",
     ...
```

Service-Type (6)

Description: This attribute specifies the type of services the link can use.

If RADIUS authenticates an incoming call using the User-Name and Password attributes, and the type of call matches the value of the Service-Type attribute, the MAX applies the attributes specified in the user profile to the call. If the type of call does not match the Service-Type attribute, the MAX rejects the call.

Usage: You can specify one of these values:

- Login (1)

  The operator can use an asynchronous Telnet connection to log into the terminal server. The MAX rejects incoming framed calls. The operator cannot use any framed protocol, but can start Telnet or raw TCP sessions.
- Framed (2)

  Incoming calls must use a framed protocol; otherwise, the MAX rejects them. Asynchronous Telnet sessions are unframed and therefore not allowed when you specify this value.
- Outbound (5)

  The user profile can be used for outgoing calls only. The MAX sends this value to the RADIUS server during an authentication request. By default, the MAX does not limit the services the link can access.

Dependencies: Keep this additional information in mind:

- Login must have an asynchronous means for reaching the MAX; that is, the MAX must have digital modems or V.110 modules, or the call must be V.120 encapsulated.

  Asynchronous Telnet sessions are unframed and therefore not allowed when you set Service-Type=Framed.

Tunnel-Client-Endpoint (66)

Description: The value of the Tunnel-Client-Endpoint attribute is the address of the initiator end of the tunnel. The attribute may appear in Access-Request and Access-Accept messages to indicate the address from which a new tunnel is to be initiated. The attribute should also be included in Accounting-Request packets if those packets contain Acct-Status-Type attributes which have the value of Start or Stop.

Usage: Enter the Tunnel-Client Endpoint attribute in user profiles. The value of the attribute is a string that is dependent on the value of the Tunnel-Medium-Type attribute that is also entered in the user profile. If the value of the Tunnel-Medium-Type is IP or IP6, the string is either a fully qualified domain name or a dotted-decimal IP address. If the value of the Tunnel-Medium-Type is not IP or IP6, the value of the Tunnel-Client-Endpoint attribute in the user profile is a tag that refers to configuration data local to the RADIUS client. The configuration data must describe the interface and medium-specific address to use as the Tunnel-Client-Endpoint value.

Example:

```
Username Password = “secret value”
Tunnel-Type = L2TP : 2,
Tunnel-Medium-Type = IP : 2,
Tunnel-Client-Endpoint = “MAXaccess” : 2
```

See Also: “Tunnel -Type (64).”
“Tunnel-Medium-Type (65).”
“Tunnel-Server-Endpoint (67).”
“Tunnel-ID (68).”
“Tunnel-Password (69).”

Tunnel-ID (68)

Description: The Tunnel-ID attribute indicates the identifier assigned to the session. The attribute appears in Accounting-Request packets which contain Acct-Status-Type attributes that have values of either Start or Stop. Together, the Tunnel-ID, Tunnel-Client-Endpoint and Tunnel- Server-Endpoint attributes provide a means to uniquely identify a tunnel session for auditing purposes.

Note: The Tunnel-ID attribute is the Acct-Tunnel-Connection attribute that is described in the “RADIUS Accounting Modifications for Tunnel Protocol Support” draft.

See Also: “Tunnel -Type (64).”
“Tunnel-Medium-Type (65).”
“Tunnel-Client-Endpoint (66).”
“Tunnel-Server-Endpoint (67).”
“Tunnel-Password (69).”

Tunneling-Protocol (127)

Description: Specifies whether a session used the ATMP tunneling protocol.

Usage: The value is ATMP if the connection used the ATMP tunneling proto col.

Example: The following is an example of a RADIUS accounting record with the Tunneling-Protocol attribute.

```
Mon Apr 21  02:41:38 1997
      User-Name = “JacobP75”
      NAS-Identifier =  1.1.1.1
      NAS-Port = 10105
      Acct-Status-Type = Stop
      Acct-Delay-Time = 0
      Acct-Session-Id = “111111111”
      Acct-Authentic = RADIUS
      Acct-Session-Time = 0
      Acct-Input-Octets = 215
      Acct-Output-Octets =  208
      Acct-Input-Packets = 10
      Acct-Output-Packets = 10
      Ascend-Disconnect-Cause = 1
      Ascend-Connect-Progress = 60
      Ascend-Data-Rate = 56000
      Ascend-PreSession-Time = 1
      Ascend-Pre-Input-Octets = 215
      Ascend-Pre-Output-Octets = 208
      Ascend-Pre-Input-Packets = 10
      Ascend-Pre-Output-Packets = 10
      Framed-Protocol = PPP
      Framed-IP-Address = 2.2.2.2
      Tunneling-Protocol = ATMP
```

Dependencies: The Tunneling-Protocol attribute is sent in Accounting-Request packets at the end of a session under the following conditions:

- The Accounting-Request packet has Acct-Status-Stop.

- The session was authenticated and encapsulated using the ATMP tunneling protocol.

Tunnel-Medium-Type (65)

Description: The Tunnel-Medium-Type attribute specifies the transport medium that can be used to create a compulsory tunnel described by the tunneling attributes in a user profile if the tunneling protocol can operate over more than one transport medium, such as L2TP. The Tunnel-Medium-Type attribute can appear in Access-Request and Access-Accept messages.

Usage: Enter Tunnel-Medium-Type as an authorization attribute in a user profile. The attribute's value is one of the one of those listed in the Assigned Numbers standard, RFC 1700 under the heading "Address Family Numbers". A relevant sample of the list includes:

- IP (IP version 4)
- IP6 (IP version 6)
- NSAP
- HDLC
- BBN 1822
- 802 (includes all 802 media plus Ethernet "canonical format")
- E.163 (POTS)
- E.164 (SMDS, Frame Relay, ATM)
- F.69 (Telex)
- X.121 (X.25, Frame Relay)
- IPX
- Appletalk
- Decnet IV
- Banyan Vines
- E.164 with NSAP format subaddresses

Example: Following is an example that displays how a Tunnel-Medium-Type attribute that has been tagged to a specific compulsory tunnel configuration might appear in a user profile.

```
Username Password = “secret value”
Tunnel-Type = L2TP : 2,
Tunnel-Medium-Type = IP : 2,
Tunnel-Server-Endpoint = “lap-aydin” : 2
```

See Also: “Tunnel -Type (64) ”
“Tunnel-Client-Endpoint (66)”
“Tunnel-Server-Endpoint (67)”
“Tunnel-ID (68)”
“Tunnel-Password (69).”

Tunnel-Password (69)

Description: The Tunnel-Password attribute contains a password that enables a user to authenticate a compulsory tunnel connection with a remote server. The attribute’s password value is hidden during transmission because it is encrypted by means of the Salted Encryption mechanism. Tunnel-Password attributes in user profiles can be tagged so that they are associated with specific compulsory tunnel configurations in the profile. The Tunnel-Password attribute can only appear in an Access-Accept message.

Note: Salted Encryption is described in a draft called “draft-ietf-radius-saltencrypt-00.txt”. The Salted Encryption mechanism permits any RADIUS attribute to be encrypted. Attributes that can encrypted by the mechanism are tagged in the RADIUS dictionary by the addition of the SALTED flag at the end of the attribute’s dictionary entry.

The Salted Encryption mechanism adds a unique two-octet Salt value to each attribute to be encrypted. The Salt is concatenated with the secret shared by the RADIUS server and the remote server and the Request Authenticator from the corresponding Access-Request. The result is input to the MD5 digest which produces an initial 16-byte XOR value that is unique for each encrypted attribute in a RADIUS transaction. The initial and subsequent XOR values are used to encrypt the attribute’s payload.

Usage: Enter the Tunnel-Password attribute and its value in a user profile. The value of the attribute is a string surrounded by quotes.

Example:

```
Username Password = “secret value”
Tunnel-Type = L2TP : 2,
Tunnel-Medium-Type = IP : 2,
Tunnel-Client-Endpoint = “MAXaccess” : 2,
Tunnel-Password = “2435saltedpass7980” : 2
```

See Also: “Tunnel -Type (64).”
“Tunnel-Medium-Type (65).”
“Tunnel-Client-Endpoint (66).”
“Tunnel-Server-Endpoint (67).”
“Tunnel-ID (68).”

Tunnel-Server-Endpoint (67)

Description: The value of the Tunnel-Server-Endpoint attribute is the address of the server end of the tunnel. The attribute may be included in an Access-Request packet. It must be included in Access-Accept packets the RADIUS server sends the NAS. The attribute should also be included in Accounting-Request packets if the packets contain an Acct-Status-Type attribute which has the value of Start or Stop and the packets pertain to a tunneled session.

Usage: Enter the Tunnel-Server Endpoint attribute in user profiles. The value of the attribute is a string that is dependent on the value of the Tunnel-Medium-Type attribute that is also entered in the user profile. If the value of the Tunnel-Medium-Type is IP or IP6, then the string is either a fully qualified domain name or a dotted-decimal IP address. If the value of the Tunnel-Medium-Type is not IP or IP6, then the value of the Tunnel-Client-Endpoint attribute in the user profile is a tag that refers to configuration data local to the RADIUS client.

Example:

```
Username Password = “secret value”
Tunnel-Type = L2TP : 2,
Tunnel-Medium-Type = IP : 2,
Tunnel-Server-Endpoint = “lap-aydin” : 2
See Also: “Tunnel -Type (64).”
“Tunnel-Medium-Type (65).”
```

`“Tunnel-Client-Endpoint (66).”`

`“Tunnel-ID (68).”`

`“Tunnel-Password (69).”`

Tunnel - Type (64)

Description: Description: The Tunnel-Type attribute specifies the protocol that the tunnel initiator should use to create a tunnel. The attribute can appear in Access-Request, Access-Accept and Accounting-Request messages.

Note: A tunnel initiator is not required to implement the tunnel type that appears in an Access-Accept message. If the tunnel initiator does not know or support the tunneling protocol represented by the numeric value of the Tunnel-Type attribute that it receives, then the initiator behaves as though the message were an Access-Reject message.

Usage: The value entered for this attribute is an acronym that is associated with a protocol. Table7-7 lists the twelve Tunnel-Type attribute tunneling protocols and the acronyms that can be entered as values for the attribute

.

Table 8-8. Tunneling protocols and their values in the Tunnel-Type attribute

Tunneling protocol	Tunnel-Type value
Point-to-Point Tunneling Protocol	PPTP
Layer Two Forwarding	L2F
Layer Two Tunneling Protocol	L2TP
Ascend Tunnel Management Protocol	ATMP
Virtual Tunneling Protocol	VTP
IP authentication Header in Tunnel Mode	AH
IP-in-IP Encapsulation	IP-IP
Minimal IP-in-IP Encapsulation	MIN-IP-IP

Table 8-8. Tunneling protocols and their values in the Tunnel-Type attribute

Tunneling protocol	Tunnel-Type value
IP Encapsulating Security Payload in Tunnel Mode	ESP
Generic Routing Encapsulation	GRE
Bay Dial Virtual Services	DVS

Example: Following is an example that displays how a Tunnel-Type attribute that has been tagged to a specific compulsory tunnel configuration, might appear in a user profile.

```
Username Password = “secret value”

Tunnel-Type = L2TP : 2,

Tunnel-Medium-Type = IP : 2,

Tunnel-Server-Endpoint = “lap-aydin” : 2
```

See Also: “Tunnel-Medium-Type (65).”
“Tunnel-Client-Endpoint (66).
“Tunnel-Server-Endpoint (67).”
“Tunnel-ID (68).”
“Tunnel-Password (69).”

User-Name (1)

Description: This attribute can specify one of the following in a user profile:

- The name of the calling device or dial-in user
- The keyword DEFAULT

  If you create a profile with the username DEFAULT and make that profile the *last profile* of the users file, the RADIUS server will use that profile to determine what to do with users who are not contained in the users file. You can configure only one DEFAULT profile in the users file.
- The incoming phone number (for CLID authentication)
- The name of a pseudo-user profile

You can set up a pseudo-user profile to configure outgoing calls, a pool of dynamic IP addresses, static IP and IPX routes, bridge entries, and the message text and host list for the terminal server interface.

Usage: Specify an alphanumeric string containing up to 252 characters. The default value is null. The username must be the first word in a user profile; you need not specify the name of the attribute.

Example: For example, consider this first line in a user profile:

```
Emma Password="pwd"
       Expiration="Sep 30 1995"
```

The username is “Emma”. The RADIUS server tests the user’s name and password against the values the user provides when making a request for access. If the RADIUS server does not find a match, it denies the request for access.

Here is an example user profile for CLID authentication using the incoming phone number as the User-Name:

```
5551212  Password="Ascend-CLID" Service-Type=Outbound
         Ascend-Require-Auth=Not-Require-Auth,
         Service-Type=Framed,
         Framed-Protocol=PPP,
         Framed-IP-Address=255.255.255.254,
         Framed-IP-Netmask=255.255.255.255,
         Ascend-Assign-IP-Pool=1,
         Ascend-Route-IP=Route-IP-Yes,
         Ascend-Idle-Limit=30
```

This example shows User-Name in a pseudo-user profile for a static route:

```
route-1 Password="ascend", Service-Type=Outbound
        Framed-Route="10.4.5.0/22 10.9.8.10 1 n inu-out"
```

SQL script for authentication and accounting table

D

Following is an SQL script for creating DBMS tables called *Authentication* and *Accounting*. The table's fields are derived from RADIUS authentication and accounting attributes. The fields are the table's column headings. You can edit the script to add or remove fields.

To create new tables, you can run the script with ISQL. RunSQL may not be used for this purpose.

```
CREATE TABLE "DBA"."Accounting"
(
"UName"                   varchar(253) NULL,
"NASIPAddr"               varchar(15)  NULL,
"NASId"                   varchar(253) NULL,
"NASPort"                 integer      NULL,
"NASPortType"             varchar(253) NULL,
"Class"                   varchar(253) NULL,
"DNIS"                    varchar(253) NULL,
"CLID"                    varchar(253) NULL,
"A_CallingSubaddr"        varchar(253) NULL,
"A_ModemPortNo"           integer      NULL,
"A_ModemSlotNo"           integer      NULL,
"A_ModemShelfNo"          integer      NULL,
"AcctStatusType"          varchar(253) NULL,
"AcctDelayTime"           integer      NULL,
"AcctInOct"                 integer      NULL,
"AcctOutOct"                integer      NULL,
"AcctSesId"                 varchar(253) NULL,
"AcctAuth"                  varchar(253) NULL,
"AcctSesTime"               integer      NULL,
"AcctInPack"                integer      NULL,
"AcctOutPack"               integer      NULL,
"AcctTermCause"             varchar(253) NULL,
"AcctMultiSesId"            varchar(253) NULL,
"AcctLinkCount"             integer      NULL,
"SvcType"                   varchar(253) NULL,
"FrmProto"                  varchar(253) NULL,
"FrmIPAddr"                 varchar(15)  NULL,
"FrmIPMask"                 varchar(15)  NULL,
"FrmRouting"                varchar(253) NULL,
"FrmMTU"                    integer      NULL,
"FrmCmpr"                   varchar(253) NULL,
"FrmRoute"                  varchar(253) NULL,
"FrmIPXNet"                 integer      NULL,
"FilterId"                  varchar(253) NULL,
"LoginIPHost"               varchar(253) NULL,
"LoginSvc"                  varchar(253) NULL,
"LoginTCPPort"              integer      NULL,
"CallbkNum"                 varchar(253) NULL,
"CallbkId"                  varchar(253) NULL,
"A_MultilinkId"             integer      NULL,
"A_NumInMultilink"          integer      NULL,
"A_FirstDest"               varchar(253) NULL,
"A_PreInOct"                integer      NULL,
"A_PreOutOct"               integer      NULL,
"A_PreInPkts"           integer        NULL,
"A_PreOutPkts"          integer        NULL,
"A_MaxTime"             integer        NULL,
"A_DisconCause"         integer        NULL,
"A_ConnProg"            integer        NULL,
"A_DataRate"            integer        NULL,
"A_PreSesTime"          integer        NULL,
"A_SesSvrKey"           varchar(253) NULL,
"A_TunProto"            varchar(253) NULL,
"A_PrimHomeAgt"         varchar(253) NULL,
"A_SecHomeAgt"          varchar(253) NULL,
"A_HomeAgentIPAddr"     varchar(15)  NULL,
"A_HomeAgentPass"       varchar(253) NULL,
"A_HomeNetName"         varchar(253) NULL,
"A_HomeAgtUDPPort"      integer        NULL,
"TunType"               varchar(253) NULL,
"TunMedType"            varchar(253) NULL,
"TunClEnd"              varchar(253) NULL,
"TunSrvEnd"             varchar(253) NULL,
"TunId"                 varchar(253) NULL,
"TunPass"               varchar(253) NULL,
"ConnInfo"              varchar(253) NULL,
"A_RedirectNum"         varchar(253) NULL,
"A_EventType"           varchar(253) NULL,
"A_NumSessions"         varchar(253) NULL,
"Start_Time"            varchar(25)  NULL,
"Stop_Time"             varchar(25)  NULL,
"A_XmitRate"            integer        NULL,
"ProxyState"            varchar(253) NULL,
"SesTtl"                integer        NULL,
"IdleTtl"                  integer      NULL,
"TermAct"                  varchar(253) NULL,
"LoginLATSvc"              varchar(253) NULL,
"LoginLATNode"             varchar(253) NULL,
"LoginLATGrp"              varchar(253) NULL,
"LoginLATPort"             varchar(253) NULL,
"FrmATLink"                integer      NULL,
"FrmATNet"                 integer      NULL,
"FrmATZone"                varchar(253) NULL,
"PortLmt"                  integer      NULL
```

```
CREATE  TABLE "DBA"."Authentication" (
"UName"                    varchar(253) NULL UNIQUE,
"NASIPAddr"                varchar(15)  NULL,
"NASPort"                  integer      NULL,
"SvcType"                  integer      NULL,
"FrmProto"                 integer      NULL,
"FrmIPAddr"                varchar(15)  NULL,
"FrmIPMask"                varchar(15)  NULL,
"FrmRouting"               integer      NULL,
"FilterId"                 varchar(253) NULL,
"FrmMTU"                   integer      NULL,
"FrmCmpr"                  integer      NULL,
"LoginIPHost"              varchar(253) NULL,
"LoginSvc"                 integer      NULL,
"LoginTCPPort"             integer      NULL,
"ReplyMsg"                 varchar(253) NULL,
"CallbkNum"                varchar(253) NULL,
"CallbkId"                 varchar(253) NULL,
"A_EnblPass"               integer      NULL,
 "FrmRoute"                varchar(253) NULL,
 "FrmIPXNet"               integer      NULL,
 "Class"                   varchar(253) NULL,
 "SesTtl"                  integer      NULL,
 "IdleTtl"                 integer      NULL,
 "TermAct"                 integer      NULL,
 "DNIS"                    varchar(253) NULL,
 "CLID"                    varchar(253) NULL,
 "NASId"                   varchar(253) NULL,
 "FrmATLink"               integer      NULL,
 "FrmATNet"                integer      NULL,
 "FrmATZone"                 varchar(253) NULL,
 "NASPortType"               integer      NULL,
 "PortLmt"                   integer      NULL,
 "TunType"                   integer      NULL,
 "TunMedType"                integer      NULL,
 "TunClEnd"                  varchar(253) NULL,
 "TunId"                     varchar(253) NULL,
 "TunPass"                   varchar(253) NULL,
 "Prompt"                    integer      NULL,
 "Comment"                   varchar(253) NULL,
 "AuthType"                  integer      NULL,
 "Prohibit"                  integer      NULL,
 "UCategory"                 varchar(253) NULL,
 "GrpName"                   varchar(253) NULL,
 "EncrPass"                  varchar(253) NULL,
 "Expiration"                DATE         NULL,
 "Passwd"                    varchar(253) NULL,
 "SesLmt"                    integer      NULL,
 "SrvName"                   varchar(253) NULL,
 "FrmIPAddrPool"             varchar(253) NULL,
 "TemplUsr"                  varchar(253) NULL,
 "GrpLmtName"                varchar(253) NULL,
 "GrpSesLmt"                 integer      NULL,
 "TimeOfDay"                 varchar(253) NULL,
 "A_IPTOS"                   integer      NULL,
 "A_IPTOSPrec"               integer      NULL,
 "A_IPTOSAppTo"              integer      NULL,
 "A_Filter"                  varchar(253) NULL,
 "A_TelnetProfile"           varchar(253) NULL,
 "A_DslRateType"             integer       NULL,
 "A_RedirectNum"          varchar(253) NULL,
 "A_ATMVpi"               integer      NULL,
 "A_ATMVci"               integer      NULL,
 "A_SrcIPCheck"           integer      NULL,
 "A_DslRateMode"          integer      NULL,
 "A_DslUpstrLim"          integer      NULL,
 "A_DslDownstLim"         integer      NULL,
 "A_DslCIRRcvLim"         integer      NULL,
 "A_DslCIRXmitLim"        integer      NULL,
 "A_VRtrName"             varchar(253) NULL,
 "A_SrcAuth"              varchar(253) NULL,
 "A_PrvRoute"             varchar(253) NULL,
 "A_NumPlanId"            integer      NULL,
 "A_FRLinkStatusDLCI"     integer      NULL,
 "A_CallingSubaddr"       varchar(253) NULL,
 "A_CallbkDelay"          integer      NULL,
 "A_EndpointDisc"         varchar(253) NULL,
 "A_RemoteFW"             varchar(253) NULL,
 "A_MulticastGLeaveDelay" integer      NULL,
 "A_CBCPEnable"           integer      NULL,
 "A_CBCPMode"             integer      NULL,
 "A_CBCPDelay"            integer      NULL,
 "A_CBCPTrunkGrp"         integer      NULL,
 "A_ATRoute"              varchar(253) NULL,
 "A_ATPeerMode"           integer      NULL,
 "A_RouteAT"              integer      NULL,
 "A_FCPParam"             varchar(253) NULL,
 "A_ModemPortNo"          integer      NULL,
 "A_ModemSlotNo"          integer      NULL,
 "A_ModemShelfNo"         integer      NULL,
 "A_CallAttemptLim"       integer        NULL,
 "A_CallBlkDur"           integer        NULL,
 "A_MaxCallDur"           integer        NULL,
 "A_RoutePref"            integer        NULL,
 "A_TunProto"             integer        NULL,
 "A_SharedProfEnable"     integer        NULL,
 "A_PrimHomeAgt"          varchar(253) NULL,
 "A_SecHomeAgt"           varchar(253) NULL,
 "A_DialoutAllowed"       integer        NULL,
 "A_ClientGateway"        varchar(253) NULL,
 "A_BACPEnable"           integer        NULL,
 "A_DHCPMaxLeases"        integer        NULL,
 "A_CliPrimDNS"           varchar(253) NULL,
 "A_CliSecDNS"            varchar(253) NULL,
 "A_CliAssignDNS"         integer        NULL,
 "A_UAcctType"            integer        NULL,
 "A_UAcctHost"            varchar(253) NULL,
 "A_UAcctPort"            integer        NULL,
 "A_UAcctKey"             varchar(253) NULL,
 "A_UAcctBase"            integer        NULL,
 "A_UAcctTime"            integer        NULL,
 "A_AssignIPClient"       varchar(253) NULL,
 "A_AssignIPSrv"          varchar(253) NULL,
 "A_AssignIPGlobalPool"   varchar(253) NULL,
 "A_DHCPReply"            integer        NULL,
 "A_DHCPPoolNum"          integer        NULL,
 "A_ExpectCallbk"         integer        NULL,
 "A_SesSvrKey"            varchar(253) NULL,
 "A_MulticastRateLim"     integer        NULL,
 "A_IFMask"               varchar(15)  NULL,
 "A_RemoteAddr"          varchar(253) NULL,
 "A_MulticastCli"        integer      NULL,
 "A_FRCircName"          varchar(253) NULL,
 "A_FRLinkUp"            integer      NULL,
 "A_FRNailedGrp"         integer      NULL,
 "A_FRType"              integer      NULL,
 "A_FRLinkMgt"           integer      NULL,
 "A_FRN391"              integer      NULL,
 "A_FRDCEN392"           integer      NULL,
 "A_FRDTEN392"           integer      NULL,
 "A_FRDCEN393"           integer      NULL,
 "A_FRDTEN393"           integer      NULL,
 "A_FRT391"              integer      NULL,
 "A_FRT392"              integer      NULL,
 "A_BridgeAddr"          varchar(253) NULL,
 "A_TSIdleLim"           integer      NULL,
 "A_TSIdleMode"          integer      NULL,
 "A_DBAMon"              integer      NULL,
 "A_BaseChanCount"       integer      NULL,
 "A_MinChan"             integer      NULL,
 "A_IPXRoute"            varchar(253) NULL,
 "A_FT1Caller"           integer      NULL,
 "A_Backup"              varchar(253) NULL,
 "A_CallType"            integer      NULL,
 "A_Grp"                 varchar(253) NULL,
 "A_FRDLCI"              integer      NULL,
 "A_FRProfName"          varchar(253) NULL,
 "A_IPXNodeAddr"         varchar(253) NULL,
 "A_HomeAgentIPAddr"     varchar(15)  NULL,
 "A_HomeAgentPass"       varchar(253) NULL,
 "A_HomeNetName"         varchar(253) NULL,
 "A_HomeAgtUDPPort"      integer      NULL,
 "A_MaxTime"             integer      NULL,
 "A_PreSesTime"          integer      NULL,
 "A_TokenIdle"           integer      NULL,
 "A_TokenImm"            integer      NULL,
 "A_ReqAuth"             integer      NULL,
 "A_AuthenAlias"         varchar(253) NULL,
 "A_TokenExpiry"         integer      NULL,
 "A_MenuSel"             varchar(253) NULL,
 "A_MenuItem"            varchar(253) NULL,
 "A_PWWarntime"          integer      NULL,
 "A_PWLifetime"          integer      NULL,
 "A_IPDirect"            varchar(253) NULL,
 "A_PPPVJSlotComp"       integer      NULL,
 "A_PPPVJ1172"           integer      NULL,
 "A_PPPAsyncMap"         integer      NULL,
 "A_SndSecret"           varchar(253) NULL,
 "A_RcvSecret"           varchar(253) NULL,
 "A_IPXPeerMode"         integer      NULL,
 "A_IPPoolDef"           varchar(253) NULL,
 "A_AsgnIPPool"          integer      NULL,
 "A_FRDir"               integer      NULL,
 "A_FRDirProfile"        varchar(253) NULL,
 "A_FRDirDLCI"           integer      NULL,
 "A_HandleIPX"           integer      NULL,
 "A_Netwarettl"          integer      NULL,
 "A_IPXAlias"            integer      NULL,
 "A_Metric"              integer      NULL,
 "A_PRINumType"          integer      NULL,
 "A_DialNum"                varchar(253) NULL,
 "A_RouteIP"                integer      NULL,
 "A_RouteIPX"               integer      NULL,
 "A_Bridge"                 integer      NULL,
 "A_SndAuth"                integer      NULL,
 "A_SndPasswd"              varchar(253) NULL,
 "A_LinkCompr"              integer      NULL,
 "A_TargetUtil"             integer      NULL,
 "A_MaxChan"                integer      NULL,
 "A_IncChanCount"           integer      NULL,
 "A_DecChanCount"           integer      NULL,
 "A_SecOfHist"              integer      NULL,
 "A_HistWeighType"          integer      NULL,
 "A_AddSec"                 integer      NULL,
 "A_RemoveSec"              integer      NULL,
 "A_IdleLmt"                integer      NULL,
 "A_PreemptLmt"             integer      NULL,
 "A_Callbk"                 integer      NULL,
 "A_DataSvc"                integer      NULL,
 "A_Force56"                integer      NULL,
 "A_BillNum"                varchar(253) NULL,
 "A_CallByCall"             integer      NULL,
 "A_TransitNum"             varchar(253) NULL,
 "A_HostInfo"               varchar(253) NULL,
 "A_PPPAddr"                varchar(15)  NULL,
 "A_MPPIdlePercent"         integer      NULL,
 "A_DataFilter"             varchar(254) NULL,
 "A_CallFilter"             varchar(254) NULL,
 "B_Filter"                 varchar(253) NULL,
 "B_CLICmd"                 varchar(253) NULL,
 "B_CLIFilter"           varchar(253)  NULL,
 "B_HostRestrict"        varchar(253)  NULL,
 "B_HostAllow"            varchar(253) NULL,
 "B_ProductName"          varchar(253) NULL,
 "B_SWVer"                varchar(253) NULL,
 "B_LocalIPAddr"          varchar(15)  NULL,
 "B_CallbkPortlist"       integer      NULL,
 "B_SecProfId"            integer      NULL,
 "B_TunAuthType"          integer      NULL,
 "B_TunAuthMode"          integer      NULL,
 "B_AuthenSrvs"           varchar(253) NULL,
 "B_AcctSrvs"             varchar(253) NULL,
 "B_UsrSrvLoc"            integer      NULL,
 "B_LocalUsrName"         varchar(253) NULL,
 "B_SysDiscReason"        integer      NULL,
 "B_ModemDiscReason"      integer      NULL,
 "B_DisconnReason"        integer      NULL,
 "B_AddrReslnProto"       integer      NULL,
 "B_AddrReslnSrvs"        varchar(253) NULL,
 "B_DomainName"           varchar(253) NULL,
 "B_XmitSpeed"            integer      NULL,
 "B_RcvSpeed"             integer      NULL,
 "LoginLATSvc"            varchar(253) NULL,
 "LoginLATNode"           varchar(253) NULL,
 "LoginLATGrp"            varchar(253) NULL,
 "LoginLATPort"           varchar(253) NULL,
);
```

radodbc.map file

E

The radodbc.map file provides the cross reference to the SQL scripts for authentication and accounting tables for mapping the columns to the appropriate attributes. Details of the radodbc.map file follow:

```
User-Name                       UName
User-Password                   UPass
CHAP-Password                   CHAPPass
NAS-IP-Address                  NASIPAddr
NAS-Port                        NASPort
Service-Type                    SvcType
Framed-Protocol                 FrmProto
Framed-IP-Address               FrmIPAddr
Framed-IP-Netmask               FrmIPMask
Framed-Routing                  FrmRouting
Filter-Id                       FilterId
Framed-MTU                      FrmMTU
Framed-Compression              FrmCmpr
Login-IP-Host                   LoginIPHost
Login-Service                   LoginSvc
Login-TCP-Port                  LoginTCPPort
Old-Password                    OldPass
Reply-Message                   ReplyMsg
```

```
Callback-Number                 CallbkNum
Callback-Id                     CallbkId
Ascend-Enable-Password          A_EnblPass
Framed-Route                    FrmRoute
Framed-IPX-Network              FrmIPXNet
State                           State
Class                           Class
Vendor-Specific                 VSA
Session-Timeout                 SesTtl
Idle-Timeout                    IdleTtl
Termination-Action              TermAct
Called-Station-Id               DNIS
Calling-Station-Id              CLID
NAS-Identifier                  NASId
Proxy-State                     ProxyState
Login-LAT-Service               LoginLATSvc
Login-LAT-Node                  LoginLATNode
Login-LAT-Group                 LoginLATGrp
Framed-AppleTalk-Link           FrmATLink
Framed-AppleTalk-Network        FrmATNet
Framed-AppleTalk-Zone           FrmATZone
CHAP-Challenge                  CHAPChal
NAS-Port-Type                   NASPortType
Port-Limit                      PortLmt
Login-LAT-Port                  LoginLATPort
Tunnel-Type                     TunType
Tunnel-Medium-Type              TunMedType
Tunnel-Client-Endpoint          TunClEnd
Tunnel-Server-Endpoint          TunSrvEnd
Tunnel-ID                       TunId
```

```
Tunnel-Password                 TunPass
Prompt                          Prompt
Connect-Info                    ConnInfo
Reply-If-Ack-Message            ReplyIfAck
LAS-Start-Time                  LASStartTime
LAS-Code                        LASCode
LAS-Duration                    LASDuration
Local-Duration                  LocalDuration
Service-Class                   SvcClass
Port-Entry                      PortEntry
Proxy-Action                    ProxyAct
Token                           Token
Huntgroup-Name                  HtgrpName
User-Id                         UId
User-Realm                      URealm
Comment                         Comment
Xvalue                          Xvalue
Xstring                         Xstring
Authentication-Type             AuthType
Prohibit                        Prohibit
User-Category                   UCategory
Group-Name                      GrpName
Encrypted-Password              EncrPass
Expiration                      Expiration
Password                        Passwd
Session-Limit                   SesLmt
Server-Name                     SrvName
Framed-IP-Address-Pool-Name     FrmIPAddrPool
Template-User                   TemplUsr
Group-Limit-Name                GrpLmtName
```

```
Group-Session-Limit             GrpSesLmt
Time-Of-Day                     TimeOfDay
Ascend-IP-TOS                   A_IPTOS
Ascend-IP-TOS-Precedence        A_IPTOSPrec
Ascend-IP-TOS-Apply-To          A_IPTOSAppTo
Ascend-Filter                   A_Filter
Ascend-Telnet-Profile           A_TelnetProfile
Ascend-Dsl-Rate-Type            A_DslRateType
Ascend-Redirect-Number          A_RedirectNum
Ascend-ATM-Vpi                  A_ATMVpi
Ascend-ATM-Vci                  A_ATMVci
Ascend-Source-IP-Check          A_SrcIPCheck
Ascend-Dsl-Rate-Mode            A_DslRateMode
Ascend-Dsl-Upstream-Limit       A_DslUpstrLim
Ascend-Dsl-Downstream-Limit     A_DslDownstLim
Ascend-Dsl-CIR-Recv-Limit       A_DslCIRRcvLim
Ascend-Dsl-CIR-Xmit-Limit       A_DslCIRXmitLim
Ascend-VRouter-Name             A_VRtrName
Ascend-Source-Auth              A_SrcAuth
Ascend-Private-Route            A_PrvRoute
Ascend-Numbering-Plan-ID        A_NumPlanId
Ascend-FR-Link-Status-DLCI      A_FRLinkStatusDLCI
Ascend-Calling-Subaddress       A_CallingSubaddr
Ascend-Callback-Delay           A_CallbkDelay
Ascend-Endpoint-Disc            A_EndpointDisc
Ascend-Remote-FW                A_RemoteFW
Ascend-Multicast-GLeave-Delay   A_MulticastGLeaveDelay
Ascend-CBCP-Enable              A_CBCPEnable
Ascend-CBCP-Mode                A_CBCPMode
Ascend-CBCP-Delay               A_CBCPDelay
```

```
Ascend-CBCP-Trunk-Group         A_CBCPTrunkGrp
Ascend-Appletalk-Route          A_ATRoute
Ascend-Appletalk-Peer-Mode      A_ATPeerMode
Ascend-Route-Appletalk          A_RouteAT
Ascend-FCP-Parameter            A_FCPParam
Ascend-Modem-PortNo             A_ModemPortNo
Ascend-Modem-SlotNo             A_ModemSlotNo
Ascend-Modem-ShelfNo            A_ModemShelfNo
Ascend-Call-Attempt-Limit       A_CallAttemptLim
Ascend-Call-Block-Duration      A_CallBlkDur
Ascend-Maximum-Call-Duration    A_MaxCallDur
Ascend-Route-Preference         A_RoutePref
Ascend-Tunneling-Protocol       A_TunProto
Ascend-Shared-Profile-Enable    A_SharedProfEnable
Ascend-Primary-Home-Agent       A_PrimHomeAgt
Ascend-Secondary-Home-Agent     A_SecHomeAgt
Ascend-Dialout-Allowed          A_DialoutAllowed
Ascend-Client-Gateway           A_ClientGateway
Ascend-BACP-Enable              A_BACPEnable
Ascend-DHCP-Maximum-Leases      A_DHCPMaxLeases
Ascend-Client-Primary-DNS       A_CliPrimDNS
Ascend-Client-Secondary-DNS     A_CliSecDNS
Ascend-Client-Assign-DNS        A_CliAssignDNS
Ascend-User-Acct-Type           A_UAcctType
Ascend-User-Acct-Host           A_UAcctHost
Ascend-User-Acct-Port           A_UAcctPort
Ascend-User-Acct-Key            A_UAcctKey
Ascend-User-Acct-Base           A_UAcctBase
Ascend-User-Acct-Time           A_UAcctTime
Ascend-Assign-IP-Client         A_AssignIPClient
```

```
Ascend-Assign-IP-Server         A_AssignIPSrv
Ascend-Assign-IP-Global-Pool    A_AssignIPGlobalPool
Ascend-DHCP-Reply               A_DHCPReply
Ascend-DHCP-Pool-Number         A_DHCPPoolNum
Ascend-Expect-Callback          A_ExpectCallbk
Ascend-Event-Type               A_EventType
Ascend-Session-Svr-Key          A_SesSvrKey
Ascend-Multicast-Rate-Limit     A_MulticastRateLim
Ascend-IF-Netmask               A_IFMask
Ascend-Remote-Addr              A_RemoteAddr
Ascend-Multicast-Client         A_MulticastCli
Ascend-FR-Circuit-Name          A_FRCircName
Ascend-FR-LinkUp                A_FRLinkUp
Ascend-FR-Nailed-Grp            A_FRNailedGrp
Ascend-FR-Type                  A_FRType
Ascend-FR-Link-Mgt              A_FRLinkMgt
Ascend-FR-N391                  A_FRN391
Ascend-FR-DCE-N392              A_FRDCEN392
Ascend-FR-DTE-N392              A_FRDTEN392
Ascend-FR-DCE-N393              A_FRDCEN393
Ascend-FR-DTE-N393              A_FRDTEN393
Ascend-FR-T391                  A_FRT391
Ascend-FR-T392                  A_FRT392
Ascend-Bridge-Address           A_BridgeAddr
Ascend-TS-Idle-Limit            A_TSIdleLim
Ascend-TS-Idle-Mode             A_TSIdleMode
Ascend-DBA-Monitor              A_DBAMon
Ascend-Base-Channel-Count       A_BaseChanCount
Ascend-Minimum-Channels         A_MinChan
Ascend-IPX-Route                A_IPXRoute
```

```
Ascend-FT1-Caller               A_FT1Caller
Ascend-Backup                   A_Backup
Ascend-Call-Type                A_CallType
Ascend-Group                    A_Grp
Ascend-FR-DLCI                  A_FRDLCI
Ascend-FR-Profile-Name          A_FRProfName
Ascend-Ara-PW                   A_AraPW
Ascend-IPX-Node-Addr            A_IPXNodeAddr
Ascend-Home-Agent-IP-Addr       A_HomeAgentIPAddr
Ascend-Home-Agent-Password      A_HomeAgentPass
Ascend-Home-Network-Name        A_HomeNetName
Ascend-Home-Agent-UDP-Port      A_HomeAgtUDPPort
Ascend-Multilink-ID             A_MultilinkId
Ascend-Num-In-Multilink         A_NumInMultilink
Ascend-First-Dest               A_FirstDest
Ascend-Pre-Input-Octets         A_PreInOct
Ascend-Pre-Output-Octets        A_PreOutOct
Ascend-Pre-Input-Packets        A_PreInPkts
Ascend-Pre-Output-Packets       A_PreOutPkts
Ascend-Maximum-Time             A_MaxTime
Ascend-Disconnect-Cause         A_DisconCause
Ascend-Connect-Progress         A_ConnProg
Ascend-Data-Rate                A_DataRate
Ascend-PreSession-Time          A_PreSesTime
Ascend-Token-Idle               A_TokenIdle
Ascend-Token-Immediate          A_TokenImm
Ascend-Require-Auth             A_ReqAuth
Ascend-Number-Sessions          A_NumSessions
Ascend-Authen-Alias             A_AuthenAlias
Ascend-Token-Expiry             A_TokenExpiry
```

```
Ascend-Menu-Selector            A_MenuSel
Ascend-Menu-Item                A_MenuItem
Ascend-PW-Warntime              A_PWWarntime
Ascend-PW-Lifetime              A_PWLifetime
Ascend-IP-Direct                A_IPDirect
Ascend-PPP-VJ-Slot-Comp         A_PPPVJSlotComp
Ascend-PPP-VJ-1172              A_PPPVJ1172
Ascend-PPP-Async-Map            A_PPPAsyncMap
Ascend-Third-Prompt             A_ThirdPrompt
Ascend-Send-Secret              A_SndSecret
Ascend-Receive-Secret           A_RcvSecret
Ascend-IPX-Peer-Mode            A_IPXPeerMode
Ascend-IP-Pool-Definition       A_IPPoolDef
Ascend-Assign-IP-Pool           A_AsgnIPPool
Ascend-FR-Direct                A_FRDir
Ascend-FR-Direct-Profile        A_FRDirProfile
Ascend-FR-Direct-DLCI           A_FRDirDLCI
Ascend-Handle-IPX               A_HandleIPX
Ascend-Netware-timeout          A_Netwarettl
Ascend-IPX-Alias                A_IPXAlias
Ascend-Metric                   A_Metric
Ascend-PRI-Number-Type          A_PRINumType
Ascend-Dial-Number              A_DialNum
Ascend-Route-IP                  A_RouteIP
Ascend-Route-IPX                A_RouteIPX
Ascend-Bridge                   A_Bridge
Ascend-Send-Auth                A_SndAuth
Ascend-Send-Passwd              A_SndPasswd
Ascend-Link-Compression         A_LinkCompr
Ascend-Target-Util              A_TargetUtil
```

```
Ascend-Maximum-Channels         A_MaxChan
Ascend-Inc-Channel-Count        A_IncChanCount
Ascend-Dec-Channel-Count        A_DecChanCount
Ascend-Seconds-Of-History       A_SecOfHist
Ascend-History-Weigh-Type       A_HistWeighType
Ascend-Add-Seconds              A_AddSec
Ascend-Remove-Seconds           A_RemoveSec
Ascend-Data-Filter              A_DataFilter
Ascend-Call-Filter              A_CallFilter
Ascend-Idle-Limit               A_IdleLmt
Ascend-Preempt-Limit            A_PreemptLmt
Ascend-Callback                 A_Callbk
Ascend-Data-Svc                 A_DataSvc
Ascend-Force-56                 A_Force56
Ascend-Billing-Number           A_BillNum
Ascend-Call-By-Call             A_CallByCall
Ascend-Transit-Number           A_TransitNum
Ascend-Host-Info                A_HostInfo
Ascend-PPP-Address              A_PPPAddr
Ascend-MPP-Idle-Percent         A_MPPIdlePercent
Ascend-Xmit-Rate                A_XmitRate
Annex-Filter                    B_Filter
Annex-CLI-Command               B_CLICmd
Annex-CLI-Filter                B_CLIFilter
Annex-Host-Restrict             B_HostRestrict
Annex-Host-Allow                B_HostAllow
Annex-Product-Name              B_ProductName
Annex-SW-Version                B_SWVer
Annex-Local-IP-Address          B_LocalIPAddr
Annex-Callback-Portlist         B_CallbkPortlist
```

```
Annex-Sec-Profile-Index         B_SecProfId
Annex-Tunnel-Authen-Type        B_TunAuthType
Annex-Tunnel-Authen-Mode        B_TunAuthMode
Annex-Authen-Servers            B_AuthenSrvs
Annex-Acct-Servers              B_AcctSrvs
Annex-User-Server-Location      B_UsrSrvLoc
Annex-Local-Username            B_LocalUsrName
Annex-System-Disc-Reason        B_SysDiscReason
Annex-Modem-Disc-Reason         B_ModemDiscReason
Annex-Disconnect-Reason         B_DisconnReason
Annex-Addr-Resolution-Protocol  B_AddrReslnProto
Annex-Addr-Resolution-Servers   B_AddrReslnSrvs
Annex-Domain-Name               B_DomainName
Annex-Transmit-Speed            B_XmitSpeed
Annex-Receive-Speed             B_RcvSpeed
Acct-Status-Type                AcctStatusType
Acct-Delay-Time                 AcctDelayTime
Acct-Input-Octets               AcctInOct
Acct-Output-Octets              AcctOutOct
Acct-Session-Id                 AcctSesId
Acct-Authentic                  AcctAuth
Acct-Session-Time               AcctSesTime
Acct-Input-Packets              AcctInPack
Acct-Output-Packets             AcctOutPack
Acct-Terminate-Cause            AcctTermCause
Acct-Multi-Session-Id           AcctMultiSesId
Acct-Link-Count                 AcctLinkCount
```

Ascend NAS Accounting codes

F

This chapter contains tables of Ascend NavisRadius Accounting codes:

NAS Accounting Disconnect Codes

Table F-1. RADIUS Accounting disconnect codes

Code	Disconnect descriptions
1	Disconnect reason is unknown.
2	Disconnect reason is unknown.
3	Call disconnected.
4	CLID authentication failed.
5	RADIUS timeout during authentication.
6	Successful authentication. Ascend unit is configured to callback user.
7	Pre-T310 Send Disc timer trigered.
9	No modem is available to accept call.
10	Modem never detected DCD.
11	Modem detected DCD, but went inactive.
12	Couldn't parse result codes.
13	Ascend unit failed to open a modem for outgoing call.
14	Ascend unit failed to open a modem for outgoing call while modemdiag diagnostic command is enabled.
20	User quit.
21	Timeout waiting for input.
22	Exiting telnet forces disconnect.
23	No IP address for switching to a framed protocol, e.g. SLIP,PPP.

Table F-1. RADIUS Accounting disconnect codes

24	Exiting raw TCP forces disconnect.
25	Exceeded login attempts.
26	Attempt to raw TCP when its disabled.
27	Saw <Ctrl-C. during logins.
28	Terminal server session cleared ungracefully.
29	User closed a virtual connect.
30	Active call for a modem outdial session closed. Originating end terminated, i.e. controlling terminal server session went down.
31	rlogin exiting.
32	Bad rlogin command line option specified.
33	Ascend unit lacks resources to process terminal server request.
35	MP+ session cleared because no null MP packets received. An Ascend unit sends (and should receive) null MP packets throughout an MP+ session.
40	LCP timed out waiting for rsp.
41	Fail to converge on LCP negotiations.
42	PAP authentication failed.
43	CHAP Authentication failed.
44	Authentication failed from remote server.
45	LCP got Terminate request from far-end while LCP was in Open state.
46	LCP got Close state from upper-layer as in Open state, i.e. normal/graceful LCP closure.
47	Closing LCP because no NCP's were opened.

Table F-1. RADIUS Accounting disconnect codes

48	For MP sessions, closing LCP; can’t determine which MP bundle to add user to.
49	For Mp sessions, closing LCP; can’t add more channels.
50	Session table is full.
51	No more resources.
52	IP address is invalid.
53	Cannot resolve hostname.
54	Bad or missing port number.
60	Host reset.
61	Connection was refused.
62	Connection timed out.
63	Connection closed by foreign host out.
64	Network unreachable.
65	Host unreachable.
66	Network admin unreachable.
67	Host admin unreachable.
68	Port unreachable.
100	Session timeout.
101	Invalid incoming user.
102	Disconnect due to callback enable.
115	Instigating call no longer active.

Table F-1. RADIUS Accounting disconnect codes

120	Protocol disabled/unsupported.
150	Disconnect requested by RADIUS server.
151	Call disconnected by local administrator.
160	Timeout, resync retries exceed maximum V110 retries (MAX_V110_RETRIES.)
170	Timeout waiting trying to authenticate.
180	User disconnectd by invoking Do Hangup from VT100 interface.
181	Call cleared by system.
185	Signal lost from far end, typically because the far end modem was turned off.
190	Resource has been quiesced.
195	Maximum duration time reached for call.
201	Ascend unit has low memory.
210	TNT modem card stops working while it has calls outstanding.
220	Ascend unit requires CBCP, but client does not support it.
230	Ascend unit deleted VROUTER.
240	Ascend unit disconnected call on the basis of LQM measurements.

NAS Accounting Progress codes

Table F-2. RADIUS Accounting Progress codes

Code	**Progress description**
1	Not applied to any call.
2	Unknown progress.
10	Ascend unit has detected and accepted call.
30	Ascend unit has assigned modem to call.
31	Modem is awaiting DCD from far-end modem.
32	Modem is awaiting result codes from far-end modem.
40	Terminal server session started.
41	Raw TCP session started.
42	Immediataed Telnet session started.
43	Connection made to raw TCP host.
44	Connection made to Telnet host.
45	Rlogin session started.
46	Connection made with Rlogin session.
47	Terminal server authentication started.
50	Modem outdial session started.
60	LAN sesssion is up.
61	Opening Link Control Protocol negotiations.
62	Opening Compression Control Protocol negotiations.

Table F-2. RADIUS Accounting Progress codes

63	Opening Internet Protocol Network Control Protocol.
64	Opening Bridging Network Control Protocol.
65	Link Control Protocol in Open state.
66	Compression Control Protocol in Open state.
67	IP NCP in Open state.
68	Bridging NCP in Open state.
69	LCP in Initial state.
70	LCP in Starting state.
71	LCP in Closed state.
72	LCP in Stopped state.
73	LCP in Closing state.
74	LCP in Stopping state.
75	LCP in Req-Sent state.
76	LCP in Ack-Rcvd state.
77	LCP in Ack-Sent state.
80	IXP NCP in Open State.
81	AT NCP in Open state.
82	BACP being opened.
83	BACP is now open.
84	CBCP being opened.

Table F-2. RADIUS Accounting Progress codes

85	CBCP is now open.
90	V110 is connected.
91	V110 in opened state.
92	V110 in carrier stat.)
93	V110 in reset state.
94	V110 in closed state.
100	Ascend unit determines that call requires callback.
101	Authentication failed.
102	Remote authentication server timed out.
120	Frame relay link is inactive. Negotiations are in progress.
121	Frame relay link is active and has end-to-end connectivity.

Index