Access Registrar Server Objects

This chapter describes the objects you use to configure and operate your Cisco Access Registrar RADIUS server.

Cisco Access Registrar is configured and operated through a set of *objects*. These objects are arranged in a hierarchy, with some of the objects containing subobjects; just as in a UNIX file system, in which directories can contain subdirectories. All of the objects, except those that are merely lists, contain properties that define the attributes or behavior of the object.

This chapter describes the Cisco Access Registrar objects:

- **Radius**—the root of the configuration hierarchy
- **UserLists**—contains individual UserLists, which in turn contain users
- **UserGroups**—contains individual UserGroups
- **Policies**—contains individual Policies
- **Clients**—contains individual Clients
- **Vendors**—contains individual Vendors
- **Scripts**—contains individual Scripts
- **Services**—contains individual Services
- **Session Managers**—contains individual Session Managers
- **Resource Managers**—contains individual Resource Managers
- **Profiles**—contains individual Profiles
- **Rules**—contains individual Rules
- **Translations**—contains individual Translations
- **TranslationGroup**—contains individual Translation Groups
- **RemoteServers**—contains individual RemoteServers
- **Advanced**—contains advanced properties, Ports, Interfaces, Reply Messages, and the Attribute dictionary.
- **Replication**—contains information about Replication

Radius

The **Radius** object is the root of the hierarchy. For each installation of the Cisco Access Registrar server, there is one instance of the **Radius** object. You reach all other objects in the hierarchy from the **Radius**.

Following is a listing of the RADIUS server object:

```
[ //localhost/Radius ]
    Name = Radius
    Description =
    Version = 1.7R0
    IncomingScript~ =
    OutgoingScript~ =
    DefaultAuthenticationService~ = local-users
    DefaultAuthorizationService~ = local-users
    DefaultAccountingService~ = local-file
    DefaultSessionService~ =
    DefaultSessionManager~ = session-mgr-1
    UserLists/
    UserGroups/
    Policies/
    Clients/
    Vendors/
    Scripts/
    Services/
    SessionManagers/
    ResourceManagers/
    Profiles/
    Rules/
    Translations/
    TranslationGroups/
    RemoteServers/
    Advanced/
    Replication/
```

Table 3-1 lists the **Radius** properties. You you can set or change Radius properties using the Cisco Access Registrar **aregcmd** commands.

> **Note** When a field is listed as required, it means a value must be supplied; that is, the value must be set. You can use the default (if it is supplied) or you can change it to something else, but you cannot unset it. You *must* supply values for the required fields and for which no defaults exist.

Table 3-1   Radius Properties

Property	Description
Name	Required; must be unique in the list of servers in the cluster
Description	Optional description of the server
Version	Required; the currently installed version of Cisco Access Registrar
IncomingScript	Optional; if there is a script, it is the first script Cisco Access Registrar runs when it receives a request from any client and/or for any service
OutgoingScript	Optional; if there is a script, it is the last script Cisco Access Registrar runs before it sends a response to any client
DefaultAuthenticationService	Optional; Cisco Access Registrar uses this property when none of the incoming scripts sets the environment dictionary variable **Authentication-Service**
DefaultAuthorizationService	Optional; Cisco Access Registrar uses this property when none of the incoming scripts sets the environment dictionary variable **Authorization-Service**

Table 3-1   Radius Properties (continued)

Property	Description
DefaultAccountingService	Optional; Cisco Access Registrar uses this property when none of the incoming scripts sets the environment dictionary variable **Accounting-Service**
DefaultSessionService	Optional; Cisco Access Registrar uses this property when none of the incoming scripts sets the environment dictionary variable **Session-Service**.
DefaultSessionManager	Optional; Cisco Access Registrar uses this property if none of the incoming scripts sets the environment dictionary variable **Session-Manager**.

The remaining Cisco Access Registrar objects are subobjects of the **Radius** object.

UserLists

The **UserLists** object contains all of the individual UserLists, which in turn, contain the specific users stored within Cisco Access Registrar. Cisco Access Registrar references each specific UserList by **name** from a Service whose type is set to **local**. When Cisco Access Registrar receives a request, it directs it to a Service. When the Service has its type property set to **local**, the Service looks up the user's entry in the specific UserList and authenticates and/or authorizes the user against that entry.

Note User names may not include the forward slash (**/**) character. If the Cisco Access Registrar server receives an access request packet with a User-Name attribute containing a forward slash character and the Cisco AR server uses an internal UserList to look up users, the server produces an error (AX_EINVAL) and may fail. If user names require a forward slash, use a script to translate the slash to an acceptable, unused character.

You can have more than one UserList in the **UserLists** object. Therefore, use the **UserLists** object to divide your user community by organization. For example, you might have separate **UserLists** objects for Company A and B, or you might have separate **UserLists** objects for different departments within a company.

Using separate **UserLists** objects allows you to have the same name in different lists. For example, if your company has three people named `Bob` and they work in different departments, you could create a UserList for each department, and each Bob could use his own name. Using UserLists lets you avoid the problem of `Bob1`, `Bob2`, and so on.

If you have more than one UserList, you can have a script Cisco Access Registrar can run in response to requests. The script chooses the Service, and the Service specifies the actual UserList which contains the user. The alternative is dynamic properties.

The subobjects are the Users listed by name. Table 3-2 lists the **UserLists** object properties.

Table 3-2   UserLists Properties

Property	Description
Name	Required. Must be unique in UserLists.
Description	Optional description of the UserList.

Users

The **Users** object contains all of the information necessary to authenticate a user or authorize a user. Users in local UserLists can have multiple profiles. Table 3-3 lists the **Users** object properties.

Table 3-3   Users Properties

Property	Description
Name	Required. Must be unique in the specific UserList.
Description	Optional description of the user.
Password	Required. The length must be between 0-253 characters.
Enabled	Required. The default is TRUE, which means the user is allowed access. Set to FALSE to cause Cisco Access Registrar to deny the user access.
Group (Overridden by User-Group)	Optional. When you set this to the name of a UserGroup, Cisco Access Registrar uses the properties specified in that UserGroup to authenticate and/or authorize the user.
BaseProfile (Overridden by User-Profile)	Optional. When you set this to the name of a Profile and the service-Type is not equal to Authenticate Only, Cisco Access Registrar adds the properties in the Profile to the Response dictionary as part of the authorization.
AuthenticationScript	Optional. When you set this property to the name of a script, you can use the script to perform additional authentication checks to determine whether to accept or reject the user.
AuthorizationScript	Optional. When you set this property to the name of a script, you can use the script to add, delete, or modify the attributes of the Response dictionary.
UserDefined1	Optional. You can use this property to store notational information, which you can then use to filter the UserList. This property also sets the environment variable for UserDefined1.

UserGroups

The **UserGroups** objects allow you to maintain common authentication and authorization attributes in one location, and then have many users reference them. By having a central location for attributes, you can make modifications in one place instead of having to make individual changes throughout your user community.

For example, you can use several **UserGroups** to separate users by the services they use, such as a group specifying PPP and another for Telnet.

Table 3-4 lists the **UserGroups** properties.

Table 3-4   UserGroups Properties

Property	Description
Name	Required. Must be unique in the **UserGroup** list.
Description	Optional description of the group.
BaseProfile	Optional. When you set this to the name of a Profile, Cisco Access Registrar adds the properties in the Profile to the response dictionary as part of the authorization.
AuthenticationScript	Optional. When you set this property to the name of a Script, you can use the Script to perform additional authentication checks to determine whether to accept or reject the user.
AuthorizationScript	Optional. When you set this property to the name of a Script, you can use the Script to add, delete, or modify the attributes of the Response dictionary.

Policies

A Policy is a set of rules applied to an Access-Request. If you are using **Policies**, the first one that must be created is SelectPolicy.

Table 3-5 lists the properties required for a given **Policy**.

Table 3-5   Policies Properties

Property	Description
Name	Required; must be unique in the **Policies** list
Description	Optional description of the Policy
Grouping	Optional grouping of rules

Clients

All NASs and proxy clients that communicate directly with Cisco Access Registrar must have an entry in the **Clients** list. This is required because NAS and proxy clients share a secret with the RADIUS server, which is used to encrypt passwords and to sign responses.

Table 3-6 lists the **Client** object properties.

Table 3-6   Client Properties

Property	Description
Name	Required and should match the Client identifier specified in the standard RADIUS attribute, **NAS-Identifier**. The name must be unique within the Clients list. For more information about standard attributes, see "RADIUS Dictionary Attributes" section on page C-1.
Description	Optional description of the client.

Table 3-6   Client Properties (continued)

Property	Description
IPAddress	Required. Must be a valid IP address and unique in the Clients list. Cisco Access Registrar uses this property to identify the Client that sent the request, either using the source IP address to identify the immediate sender and/or using the **NAS-IP-Address** attribute in the Request dictionary to identify the NAS sending the request through a proxy.
SharedSecret	Required. Must match the secret configured in the Client.
Type	Required. Accept the default, which is NAS, or set it to Proxy or NAS+Proxy.
Vendor	Optional. You can use this property when you need special processing for a specific vendor's NAS. To use this property, you must configure a **Vendor** object and include a Script. Cisco Access Registrar provides five Scripts you can use: one for Ascend, Cisco, Cabletron, Altiga, and one for USR. You can also provide your own Script.
IncomingScript	Optional. You can use this property to specify a Script you can use to determine the services to use for authentication, authorization, and/or accounting.
OutgoingScript	Optional. You can use this property to specify a Script you can use to make any Client-specific modifications when responding to a particular Client.

Vendors

The **Vendor** object provides a central location for specifying all of the request and response processing a particular NAS or Proxy vendor requires. Depending on the vendor, it may be necessary to map attributes in the request from one set to another, or to filter out certain attributes before sending the response to the client. For more information about standard RADIUS attributes, see the List of Attributes in "RADIUS Dictionary Attributes" section on page C-1.

> **Note** When you have also set **/Radius/IncomingScript**, Cisco Access Registrar runs that script before the vendor's script. Conversely, when you have set a **/Radius/Outgoing** script, Cisco Access Registrar runs the vendor's script before that script.

Table 3-7 lists the **Vendor** object properties.

Table 3-7   Vendor Properties

Property	Description
Name	Required. Must be unique in the Vendors list.
Description	Optional description of the vendor.
IncomingScript	Optional. When you specify an IncomingScript, Cisco Access Registrar runs the script on all requests from clients that specify that vendor.
OutgoingScript	Optional. When you specify an OutgoingScript, Cisco Access Registrar runs the script on all responses to the Client.

Scripts

The **Script** objects define the function Cisco Access Registrar invokes whenever the **Script** is referenced by name from other objects in the configuration.

You can write two types of scripts:

- REX (RADIUS EXtension) scripts are written in C or C++, and thus are compiled functions that reside in shared libraries

- Tcl scripts are written in Tcl, and are interpreted functions defined in source files.

Note   For more information about how to write scripts and how to incorporate them into Cisco Access Registrar, see Chapter 5, "Using Extension Points."

Table 3-8 lists the **Script** object properties.

Table 3-8   Script Properties

Property	Description
Name	Required. Must be unique in the Scripts list.
Description	Optional description of the script.
Language	Required. You must specify either **REX** or **Tcl**.
Filename	Required. You can specify either a relative or absolute path. When you specify a relative path, the path must be relative to the **$INSTALL/scripts/radius/$Language** directory. When you specify an absolute path, the server must be able to reach it.
EntryPoint	Optional. When you do not set this property, Cisco Access Registrar uses the value specified in the **Name** property.
InitEntryPoint	Optional. When you set it, it must be the name of the global symbol Cisco Access Registrar should call when it initializes the shared library at system start up, and just before it unloads the shared library.
InitEntryPointArg	Optional. When you set it, it must be the arguments to be passed to the **InitEntryPoint** in the environmental variable **Arguments**.

The **InitEntryPoint** properties allow you to perform initialization before processing and then cleanup before stopping the server. For example, when Cisco Access Registrar unloads the script (when it stops the RADIUS server) it calls the **InitEntryPoint** again to allow it to perform any clean-up operations as a result of its initialization. One use of the function might be to allow the script to close an open Accounting log file before stopping the RADIUS server.

Note   When you use Cisco Access Registrar's file service, Cisco Access Registrar automatically closes any opened files; however, if you write scripts that manipulate files, you are responsible for closing them.

Services

Cisco Access Registrar supports AAA Services (authentication, authorization, and/or accounting). In addition to the variety of built-in AAA services (specified in the **Type** property), Cisco Access Registrar also provides you with the ability to add new AAA services through custom shared libraries.

Table 3-9 lists the **Services** properties.

Table 3-9   Services Properties

Property	Description
Name	Required. Must be unique in the Services list.
Description	Optional description of the service.
Type	Required. You must set it to one of the following: **local**, **radius**, **tacacs-udp**, **ldap**, **file**, **radius-session**, **group**, or **rex**.
OutagePolicy	Required. The default is **RejectAll**. This property defines how Cisco Access Registrar handles requests if all servers listed in the **RemoteServers** properties are unavailable (that is, all remote RADIUS servers are not available). You must set it to one of the following: **AcceptAll**, **DropPacket**, or **RejectAll**.
OutageScript	Optional. If you set this property to the name of a script, Cisco Access Registrar runs it when an outage occurs. This property allows you to create a script that notifies you when the RADIUS server detects a failure.

Note, **OutagePolicy** also applies to Accounting-Requests. If an Accounting-Request is directed to an unavailable Service, then the values in Table 3-10 apply.

Table 3-10   OutagePolicy Request Packets

Value	Description	Accounting-Request Description
AcceptAll	Continues processing the packet as if the Service was successful.	The Accounting-Request will continue through the server and a response will be sent.
DropPacket	Immediately drops the packet, no further processing, and does not send any response to the client for this packet.	The packet will be discarded and it will not be processed any further.
RejectAll	Rejects the packet, but continues processing it and sends the client a reject response.	The packet will continue to flow through the server, including Session Management, if so configured, but no response will be sent. This allows you to configure the server so resources allocated by a SessionManager can be released as soon as possible, while still indicating to the client that it should keep retrying the request (with the hope the Service will be available).

Types of Services

The Service you specify determines what additional information you must supply. The following are the types with their required and optional fields.

local

Specify **local** when you want Cisco Access Registrar's RADIUS server to perform the authentication and/or authorization using a specific UserList. For more information, see the "UserLists" section on page 3-3.

radius, ldap, or tacacs-udp

Specify one of the following Services when you want to use a particular remote server for:

- **radius**—authentication and/or authorization
- **ldap**—authentication and/or authorization

> ✎
>
> **Note**   When using LDAP for authentication and a local database for authorization, ensure that the usernames in both locations are identical with regard to case sensitivity.

- **tacacs-udp**—authentication.

You must supply the information listed in Table 3-11.

Table 3-11   radius, ldap, or tacacs-udp Properties

Property	Description
MultipleServersPolicy	Required. Must be set to either **Failover** or **RoundRobin**.
	When you set it to **Failover**, Cisco Access Registrar directs requests to the first server in the list until it determines the server is off-line. At which time, Cisco Access Registrar redirects all requests to the next server in the list until it finds a server that is on-line.
	When you set it to **RoundRobin**, Cisco Access Registrar directs each request to the next server in the RemoteServers list in order to share the resource load across all of the servers listed in the RemoteServers list.
RemoteServers	Required. An indexed list from 1 to *<n>*. Each entry in the list is the name of a RemoteServer.

file

You specify the **file** Service when you want Cisco Access Registrar's RADIUS Server to perform local accounting using a specific file.

You must supply the information listed in Table 3-12.

Table 3-12   File Properties

Property	Description
FilenamePrefix	Required. A string that specifies where Cisco Access Registrar writes the account records. It must be either a relative or absolute path. When you specify a relative path, it must be relative to the **$INSTALL/logs** directory. When you specify an absolute path, the server must be able to reach it. The default is **Accounting**.
MaxFileSize	Optional. Stored as a string, but is composed of two parts, a number and a units indicator (<n> <units>) in which the unit is one of: K, Kilobyte, Kilobytes, M, Megabyte, Megabytes, G, Gigabyte, Gigabytes. The default is 10 Megabytes.
MaxFileAge	Optional. Stored as a string, but is composed of two parts, a number and a units indicator (*<n> <units>*) in which the unit is one of: H, Hour, Hours, D, Day, Days, W, Week, Weeks. The default is 1 Day.

Cisco Access Registrar opens the file when it starts the RADIUS server and closes the file when you stop the server. You can depend on Cisco Access Registrar to flush the accounting record to disk before it acknowledges the request.

Based on the maximum file size and age you have specified, Cisco Access Registrar closes the accounting file and moves it to a new name and reopens the file as a new file. The name Cisco Access Registrar gives this accounting file depends on its creation and modification dates.

- If the file was created and modified on the same date, the file name is **FileNamePrefix-***<yyyymmdd>***-***<n>***.log**. The date is displayed as year, month, day, number.

- If the file was created on one day and modified on another, the file name is **FileNamePrefix-***<yyyymmdd>***-***<yyyymmdd>***-***<n>***.log**. The dates are creation, modification, and number.

rex

Specify the **rex** service type when you want to create a custom service by using a script for authentication, authorization, and/or accounting.

You must supply the information listed in Table 3-13.

Table 3-13   rex Properties

Property	Description
Filename	Required. Must be either a relative or an absolute path to the shared library containing the Service. When the path name is relative, it must be relative to **$INSTALL/Scripts/Radius/rex**.
EntryPoint	Required. Must be set to the function's global symbol.

Property	Description
InitEntryPoint	Required. Must be the name of the global symbol Cisco Access Registrar should call when it initializes the shared library and just before it unloads the shared library. **Note** A **rex** service must have an InitEntryPoint even if the service only returns REX_OK.
InitEntryPointArgs	Optional. When you set it, it must be the arguments to be passed to the **InitEntryPoint** in the environmental variable **Arguments**.

For more information about scripting, see Chapter 5, "Using Extension Points." For more information about using the REX Attribute dictionary, see Appendix A, "Cisco Access Registrar Tcl and REX Dictionaries,".

Session Managers

You can use session management to track user sessions. The Session Managers monitor the flow of requests from each NAS and detect the session state. When requests come through to the Session Manager, it creates sessions, allocates resources from appropriate Resource Managers, and frees and deletes sessions when users log out.

The Session Manager enables you to allocate dynamic resources to users for the lifetime of their session. You can define one or more Session Managers and have each one manage the sessions for a particular group or company.

Session Managers use Resource Managers, which in turn, manage a pool of resources of a particular type.

Table 3-14 lists the Session Manager properties.

Table 3-14   Session Manager Properties

Property	Description
Name	Required. Must be unique in the Session Managers list.
Description	Optional. Description of the Session Manager.
Resource Managers	Ordered list of Resource Managers.

You can manage sessions with the two **aregcmd** session management commands: **query-sessions** and **release-sessions**. For more information about these two commands, see the "query-sessions" section on page 2-6 and the "release-sessions" section on page 2-6.

Session Creation

Cisco Access Registrar Sessions can be created by two types of RADIUS packets:

- Access-Requests
- Accounting-Requests with an **Acct-Status-Type** attribute with a value of **Start**.

This allows Cisco Access Registrar to monitor Sessions even when it is not allocating resources. For example, when Cisco Access Registrar is being used as an "Accounting-Only" server (only receiving Accounting requests), it can create a Session for each Accounting "Start" packet it successfully processes. The corresponding Accounting "Stop" request will clean up the Session. Note, if a Session already exists for that NAS/NAS-Port/User (created by an Access-Request), Cisco Access Registrar will not create a new one.

When you do not want Cisco Access Registrar to create Sessions for Accounting "Start" requests, simply set the **AllowAccountingStartToCreateSession** property on the SessionManager to FALSE.

Session Notes

Session Notes are named text messages attached to a Session and are stored with the Session data, including resources allocated for a specific user session. This data, including Session Notes, can be retrieved and viewed using the **aregcmd** command **query-sessions**.

--> **query-sessions /Radius/SessionManagers/session-mgr-2**

```
sessions for /Radius/SessionManagers/session-mgr-2:
S257 NAS: localhost, NAS-Port:1, User-Name: user1, Time: 00:00:08,
   IPX 0x1, GSL 1, USL 1, NOTES: "Date" "Today is 12/14/98.", "Requested
   IP Address" "1.2.3.4", "Framed-IP-Address" "11.21.31.4"
```

Session Notes can be created by Scripts using the Environment dictionary passed into each or by the Cisco Access Registrar server. When more than one Session Note is added, the **Session-Notes** entry should be a comma-separated list of entry names.

For a TCL script:

Step 1 The Script should create an Environment dictionary entry using the Session Note name as the entry name, and the Session Note text as the entry value. For example:

```
$environ put "Date" "Today is 12/15/98"
$environ put "Request IP Address" "1.2.3.4"
```

Step 2 The Script should create/set an Environment dictionary entry with the name **Session-Notes** with a value that contains the name of the entries created. For example:

```
$environ put "Session-Notes" "$Date, $Requested_IP_Address"
```

For a REX script:

Step 1 The Script should create an Environment dictionary entry using the Session Note name as the entry name, and the Session Note text as the entry value. For example:

```
pEnviron-->put(pEnviron, Date, "Today is 12/15/98.");
pEnviron-->put(pEnviron, Request_IP_Address, "1.2.3.4");
```

Step 2 The Script should create/set an Environment dictionary entry with the name **Session-Notes** with a value that contains the name of the first entry created. For example:

```
pEnviron-->put(pEnviron, "Session-Notes", "Date, Requested_IP_Address");
```

> **Note** Scripts creating Session Notes must be executed before the Session Management step takes place while processing a packet.

Cisco Access Registrar will automatically create a Session Note if a packet is passed to a SessionManager and it already contains a **Framed-IP-Address** attribute in the packet's Response dictionary. This IP address could come from a Profile, RemoteServer response, or from a previously executed script. For example, a Session output containing Session Notes when using the **aregcmd** command **query-session** would be as follows:

```
sessions for /Radius/SessionManagers/session-mgr-2:
  S257 NAS: localhost, NAS-Port:1, User-Name: user1, Time: 00:00:08,
  IPX 0x1, GSL 1, USL 1, NOTES: "Date" "Today is 12/14/98.", "Requested
  IP Address" "1.2.3.4", "Framed-IP-Address" "11.21.31.4"
```

Session Notes are also copied into the Environment dictionary after Session Management. The **Session-Notes** Environment dictionary entry will contain the names of all the Environment dictionary entries containing Session Notes.

Soft Group Session Limit

Two new environment variables, **Group-Session-Limit** and **Current-Group-Count** (see rex.h), are set if the group session limit resource is allocated for a packet. These variables allow a script to see how close the group is to its session limit; one way to use this information is to implement a script-based soft limit. For example, you could use the Class attribute to mark sessions that have exceeded a soft limit of 80% -- as hard coded in the script (in a Tcl script called from /Radius/OutgoingScript):

```
set softlimit [ expr 0.8 * [ $environ get Group-Session-Limit ] ]

if { [ $environ get Current-Group-Count ] < $softlimit } {

$response put Class 0

} else {

$response put Class 1

}
```

> **Note** The soft limit itself is hard coded in the script; soft limits are not directly supported in the server. The action to be taken when the soft limit is exceeded (for example, Class = 1, and then the accounting software branches on the value of Class) is also the responsibility of the script and/or external software.

Session Correlation Based on User-Defined Attributes

With the release of Cisco Access Registrar 1.7, all the session objects are maintained in one dictionary keyed by a string.

You can define the keying material to the session dictionary through a newly introduced environment variable, **Session-Key**. If the **Session-Key** is presented at the time of session manager process, it will be used as the key to the session object for this session. The **Session-Key** is of type string. By default, the **Session-Key** is not set. It's value should come from attributes in the incoming packet and is typically set by scripts. For example, CLID can be used to set the value of **Session-Key**.

Use the function UseCLIDAsSessionKey as defined in the script **rexscript.c** to specify that the **Calling-Station-Id** attribute that should be used as the session key to correlate requests for the same session. This is a typical case for 3G mobile user session correlation. You can provide your own script to define other attributes as the session key.

In the absence of the **Session-Key** variable, the key to the session will be created based on the string concatenated by the value of the **NAS-Identifier** and the **NAS-Port**.

There is a new option *with-key* available in **aregcmd** for query-sessions and release-sessions to access sessions by **Session-Key**.

Resource Managers

Resource Managers allow you to allocate dynamic resources to user sessions. The following lists the different types of Resource Managers.

- **IP-Dynamic**—manages a pool of IP addresses that allows you to dynamically allocate IP addresses from a pool of addresses
- **IP-Per-NAS-Port**—allows you to associate ports to specific IP addresses, and thus ensure each NAS port always gets the same IP address
- **IPX-Dynamic**—manages a pool of IPX network addresses
- **Subnet-Dynamic**—manages a pool of subnet addresses
- **Group-Session-Limit**—manages concurrent sessions for a group of users; that is, it keeps track of how many sessions are active and denies new sessions once the configured limit has been reached
- **User-Session-Limit**—manages per-user concurrent sessions; that is, it keeps track of how many sessions each user has and denies the user a new session once the configured limit has been reached
- **Home-Agent**—manages a pool of on-demand IP addresses
- **USR-VPN**—manages Virtual Private Networks (VPNs) that use USR NAS Clients.

Each Resource Manager is responsible for examining the request and deciding whether to allocate a resource for the user, do nothing, or cause Cisco Access Registrar to reject the request.

Table 3-15 lists the Resource Manager properties.

Table 3-15   Resource Manager Properties

Property	Description
Name	Required. Must be unique in the Resource Managers list.
Description	Optional. Description of the Resource Manger.
Type	Required. Must be either **IP-Dynamic**, **IP-Per-NAS-Port**, **IPX-Dynamic**, **Group-Session-Limit**, **Home-Agent**, **User-Session-Limit**, or **USR-VPN**.

Types of Resource Managers

A number of different types of Resource Managers exist that allow you to manage IP addresses dynamically or statically, limit sessions on a per group or per user basis, or manage a Virtual Private Network. See Appendix A, "Cisco Access Registrar Tcl and REX Dictionaries" for information on how to override these individual Resource Managers.

IP-Dynamic

IP-Dynamic allows you to manage a pool of IP addresses from which you dynamically allocate IP addresses.

When you use this Resource Manager, supply the information listed in Table 3-16.

Table 3-16   IP-Dynamic Properties

Property	Description
NetMask	Required. Must be set to a valid net mask.
IPAddresses	Required. Must be a list of IP address ranges.

IP-Per-NAS-Port

IP-Per-NAS-Port allows you to associate specific IP addresses with specific NAS ports and thus ensures each NAS port always gets the same IP address.

When you use this Resource Manager, supply the information listed in Table 3-17.

Note You must have the same number of IP addresses and ports.

Table 3-17   IP-Per-NAS-Port Properties

Property	Description
NetMask	Required. If used, must be set to a valid net mask.
NAS	Required. Must be the name of a known Client.This value must be the same as the NAS-Identifier attribute in the Access-Request packet.
IPAddresses	Required; must be a list of IP address ranges.
NASPorts	Required; a list of NAS ports.

IPX-Dynamic

An **IPX-Dynamic** Resource Manager allows you to dynamically manage a pool of IPX networks. When you use an IPX-Dynamic Resource Manager, supply the information listed in Table 3-18.

Table 3-18   IPX-Dynamic Property

Property	Description
Networks	Required. Must be a valid set of numbers which correspond to your networks.

Note You may not use IPX network number 0x0. If you attempt to configure a Resource Manager with an IPX network number of 0x0, validation will fail.

Subnet-Dynamic

A **subnet-dynamic** Resource Manager was created to support the On Demand Address Pool feature. Subnet-dynamic resource managers are used to provide pools of subnet addresses. Following is an example of the configuration of a subnet dynamic resource manager:

```
/Radius/ResourceManagers/newResourceMgr
Name = newResourceMgr
Description =
Type = subnet-dynamic
Subnet-Mask = 255.255.255.0
SubnetAddresses/
    10.1.0.0-10.1.10.0
    11.1.0.0-11.1.10.0
```

When you use this Resource Manager, supply the information listed in Table 3-19.

Table 3-19   Subnet-Dynamic Properties

Property	Description
Type	Required
Subnet mask	Required; must be set to the size of the managed subnets
SubnetAddresses	Required; must be a valid range of IP addresses

Group-Session-Limit

Group-Session-Limit allows you to manage concurrent sessions for a group of users; that is, it keeps track of how many sessions are active and denies new sessions once the configured limit has been reached.

When you use this Resource Manager, supply the information listed in Table 3-20.

Table 3-20   Group-Session-Limit Property

Property	Description
GroupSessionLimit	Required. Must be set to the maximum number of concurrent sessions for all users.

User-Session-Limit

User-Session-Limit allows you to manage per-user concurrent sessions; that is, it keeps track of how many sessions each user has and denies the user a new session once the configured limit has been reached.

When you use this Resource Manager, supply the information listed in Table 3-21.

Table 3-21   User-Session-Limit Property

Property	Description
UserSessionLimit	Required. Must be set to the maximum number of concurrent sessions for a particular user.

Home-Agent

Home-Agent is a new resource manager that supports dynamic HA assignment. You configure the home-agent resource manager with a list of IP addresses. The AR server assigns those addresses to clients whose request dictionary has the right attributes to indicate that an assignment should be done. This is similar to the **ip-dynamic** resource manager.

Unlike the **ip-dynamic** resource manager, HAs are not exclusively allocated to an individual session but are shared among a set of sessions.

Detailed configuration information for the Home-Agent resource manager is found in Chapter 10, "Wireless Support". When you use this Resource Manager, supply the information listed in Table 3-22.

Table 3-22   Home-Agent Subdirectory

Subdirectory	Description
Home-Agent-IPAddresses	A single IP address or a range of IP addresses

USR-VPN

USR-VPN allows you to set up a Virtual Private Network (VPN) using a US Robotics NAS.

When you use this Resource Manager, supply the information listed in Table 3-23.

Table 3-23   USR-VPN Properties

Property	Description
Identifier	Required. Must be set to the VPN ID the USR NAS will use to identify a VPN.
Neighbor	Optional. If set, should be the IP address of the next hop router for the VPN.
FramedRouting	Optional. If set, should be **RIP V2 Off** or **RIP V2 On** if the USR NAS is to run RIP Version 2 for the user.
Gateways	Required to set up a tunnel between the NAS and the Gateways.

Gateway Subobject

The **Gateway** subobject includes a list of names of the Frame Relay Gateways for which to encrypt the session key.

If you use this Resource Manager, supply the information listed in Table 3-24.

Table 3-24   Gateway Properties

Property	Description
Name	Required. Must be unique in the Gateways list.
Description	Optional. Description of the gateway.
IPAddress	Required. The IP address of the gateway.
SharedSecret	Required. Must match the shared secret of the gateway.
TunnelRefresh	Optional. If specified it is the number of seconds the tunnel stays active before a secure "keepalive" is exchanged between the tunnel peers in order to maintain the tunnel open.
LocationID	Optional. If specified it is a string indicating the physical location of the gateway.

Profiles

You use Profiles to group RADIUS attributes that belong together, such as attributes that are appropriate for a particular class of PPP or Telnet user. You can reference profiles by name from either the **UserGroup** or the **User** properties. Thus, if the specifications of a particular profile change, you can make the change in a single place and have it propagated throughout your user community.

Although you can use UserGroups or Profiles in a similar manner, choosing whether to use one rather than the other depends on your site. When you require some choice in determining how to authorize or authenticate a user session, then creating specific profiles, and creating a group that uses a script to choose among them is more flexible.

In such a situation, you might create a default group, and then write a script that selects the appropriate profile based on the specific request. The benefit to this technique is each user can have a single entry, and use the appropriate profile depending on the way they log in.

Table 3-25 lists the **Profile** properties.

Table 3-25   Profile Properties

Property	Description
Name	Required. Must be unique in the Profiles list.
Description	Optional. Description of the profile.
Attributes	Profiles include specific RADIUS attributes that Cisco Access Registrar returns in the Access-Accept response.

Attributes

Attributes are specific RADIUS components of requests and responses defined in the Request and Response Attribute dictionaries. Use the **aregcmd** command **set** to assign values to attributes.

For a complete list of the attributes, see Appendix C, "RADIUS Attributes." Table 3-26 lists the **Attribute** properties.

Table 3-26   Attribute Properties

Property	Description
Name=value	The attribute name is one of the attributes defined in the Attribute dictionaries. The value is appropriate for the type of attribute.

When setting a value for a STRING-type attribute such as Connect-Info (which starts with an integer), you must use the hexadecimal representation of the integer. For example, to set the attribute Connect-Info to the value 7:7, use a set command like the following:

set Connect-Info 37:3A:37

Translations

Translations add new attributes to a packet or change an existing attribute from one value to another. The **Translations** subdirectory lists all definitions of **Translations** the RADIUS server can apply to certain packets.

Under the **/Radius/Translations** directory, any translation to insert, substitute, or translate attributes can be added. The following is a sample configuration under the **/Radius/Translations** directory:

```
cd /Radius/Translations
Add T1
cd T1
Set DeleAttrs Session-Timeout,Called-Station-Id
cd Attributes
Set Calling-Station-Id 18009998888
```

DeleAttrs is the set of attributes to be deleted from the packet. Each attribute is comma separated and no spaces are allowed between attributes. All attribute value pairs under the attributes subdirectory are the attributes and values that are going to be added or translated to the packet.

Under the **/Radius/Translations/T1/Attributes** directory, inserted or translated attribute value pairs can be set. These attribute value pairs are either added to the packet or replaced with the new value.

If a translation applies to an Access-Request packet, by referencing the definition of that translation, the CAR server modifies the Request dictionary and inserts, filters and substitutes the attributes accordingly. You can set many translations for one packet and the CAR server applies these translations sequentially.

> **Note** Later translations can overwrite previous translations.

Table 3-27 lists the Translation properties.

Table 3-27   Translations Properties

Property	Description
Name	Required; must be unique in the Translations list.
Description	Optional; description of the Translation
DeleteAttrs	Optional; lists attributes to be filtered out

TranslationGroups

You can add translation groups for different user groups under **TranslationGroups**. All Translations under the Translations subdirectory are applied to those packets that fall into the groups. The groups are integrated with the CAR Rule engine.

The CAR Administrator can use any RADIUS attribute to determine the **Translation Group**. The incoming and outgoing translation group can be different translation groups. For example, you can set one translation group for incoming translations and one for outgoing translations.

Under the **/Radius/TranslationGroups** directory, translations can be grouped and applied to certain sets of packets, which are referred to in a rule. The following is a sample configuration under the **/Radius/TranslationGroups** directory:

```
cd /Radius/TranslationGroups
Add CiscoIncoming
cd CiscoIncoming
cd Translations
Set 1 T1
```

The translation group is referenced through the Cisco Access Registrar Policy Engine in the **/Radius/Rules/**<*RuleName*>**/Attributes** directory. **Incoming-Translation-Groups** are set to a translation group (for example `CiscoIncoming`) and **Outgoing-Translation-Groups** to another translation group (for example `CiscoOutgoing`). Table 3-28 lists the Translation Group properties.

Table 3-28   TranslationGroups Properties

Property	Description
Name	Required; must be unique in the Translations list.
Description	Optional; description of the Translation Group
Translations	Lists of translation

Remote Servers

You can use the **RemoteServers** object to specify the properties of the remote servers to which Services proxy requests. **RemoteServers** are referenced by name from the **RemoteServers** list in either the **radius**, **ldap** or **tacacs-udp** Services.

Table 3-29 lists the **RemoteServers** properties.

Table 3-29   RemoteServers Properties

Property	Description
Name	Required. Must be unique in the RemoteServers list.
Description	Optional. Description of the remote server.
Protocol	Required. Specifies the remote server protocol which can be **radius**, **ldap**, or **tacacs-udp**.
IPAddress	Required. This property specifies where to send the proxy request. It is the address of the remote server. You must set it to a valid IP address.
Port	Required. The port to which Cisco Access Registrar sends proxy requests. You must specify a number greater than zero. There is no default port number, you must supply the correct port number for your remote server.
ReactivateTimerInterval	Required. The amount of time (in milliseconds) to wait before retrying a remote server that was offline. You must specify a number greater than zero. The default is 300,000 (5 minutes).

Types of Protocols

The protocol you specify determines what additional information you must supply. The following are all of the protocols with their required and optional fields.

radius

radius specifies a RADIUS server.

When you specify the **radius** protocol, supply the information in Table 3-30.

Table 3-30   RADIUS Properties

Property	Description
SharedSecret	Required. The secret shared between the remote server and the RADIUS server.
IncomingScript	Optional. When set, must be the name of a known incoming script. Cisco Access Registrar runs the IncomingScript after it receives the response.
OutgoingScript	Optional. When set, must be the name of a known outgoing script. Cisco Access Registrar runs the OutgoingScript just before it sends the proxy request to the remote server.
Vendor	Optional. When set, must be the name of a known Vendor.
MaxTries	Required. The number of times to send a proxy request to a remote server before deciding the server is off-line. You must specify a number greater than zero. The default is 3.
InitialTimeout	Required. Represents the number of milliseconds used as a timeout for the first attempt to send a specific packet to a remote server. For each successive retry on the same packet, the previous timeout value used is doubled. You must specify a number greater than zero. The default value is 2000 (or 2 seconds).

ldap

ldap specifies an LDAP server. When you specify the **ldap** protocol, supply the information listed in Table 3-31.

Table 3-31   ldap Properties

Property	Description
Timeout	Required. The default is 15. The timeout property indicates how many seconds the RADIUS server will wait for a response from the LDAP server. **Note** Use InitialTimeout from above as a template, except this is timeout is specified in seconds.
HostName	Required. The LDAP server's host name or IP address.
BindName	Optional. The distinguished name (dn) to use when establishing a connection between the LDAP and RADIUS servers.
BindPassword	Optional. The password associated with the **BindName**.

Table 3-31   ldap Properties (continued)

Property	Description
SearchPath (Overridden by Search-Path environment variable)	Required. The path that indicates where in the LDAP database to start the search for user information.
Filter	Required. This specifies the search filter Cisco Access Registrar uses when querying the LDAP server for user information. When you configure this property, use the notation "%s" to indicate where the user ID should be inserted. For example, a typical value for this property is "(uid=%s)," which means that when querying for information about user `joe`, use the filter `uid=joe`.
UserPasswordAttribute	Required. This specifies which LDAP field the RADIUS server should check for the user's password.
LimitOutstandingRequests	Required. The default is FALSE. Cisco Access Registrar uses this property in conjunction with the **MaxOutstandingRequests** property to tune the RADIUS server's use of the LDAP server. When you set this property to TRUE, the number of outstanding requests for this RemoteServer is limited to the value you specified in **MaxOutstandingRequests**. When the number of requests exceeds this number, Cisco Access Registrar queues the remaining requests, and sends them as soon as the number of outstanding requests drops to this number.
MaxOutstandingRequests	Required when you have set the **LimitOutstandingRequests** to TRUE. The number you specify, which must be greater than zero, determines the maximum number of outstanding requests allowed for this remote server.
MaxReferrals	Required. Must be a number equal to or greater than zero. This property indicates how many referrals are allowed when looking up user information. When you set this property to zero, no referrals are allowed. Cisco Access Registrar manages referrals by allowing the RADIUS server's administrator to indicate an LDAP "referral attribute," which may or may not appear in the user information returned from an LDAP query. When this information is returned from a query, Cisco Access Registrar assumes it is a referral and initiates another query based on the referral. Referrals can also contain referrals. **Note** This is an LDAP v2 referral property.
ReferralAttribute	Required when you have specified a **MaxReferrals** value. This property specifies which LDAP attribute, returned from an LDAP search, to check for referral information. **Note** This is an LDAP v2 referral property.
ReferralFilter	Required when you have specified a **MaxReferral** value. This is the filter Cisco Access Registrar uses when processing referrals. When checking referrals, the information Cisco Access Registrar finds in the referral itself is considered to be the search path and this property provides the filter. The syntax is the same as that of the **Filter** property. **Note** This is an LDAP v2 referral property.
PasswordEncryptionStyle	The default is **None**. You can also specify **crypt**, **dynamic**, **SHA-1**, and **SSHA-1**.

Table 3-31   ldap Properties (continued)

Property	Description
LDAPToRadiusMappings	A list of name/value pairs in which the name is the name of the **ldap** attribute to retrieve from the user record, and the value is the name of the RADIUS attribute to set to the value of the **ldap** attribute retrieved. For example, when the **LDAPToRadiusMappings** has the entry: **FramedIPAddress = Framed-IP-Address**, the RemoteServer retrieves the **FramedIPAddress** attribute from the **ldap** user entry for the specified user, uses the value returned, and sets the Response variable **Framed-IP-Address** to that value.
LDAPToEnvironmentMappings	A list of name/value pairs in which the name is the name of the **ldap** attribute to retrieve from the user record, and the value is the name of the Environment variable to set to the value of the **ldap** attribute retrieved. For example, when the **LDAPToEnvironmentMappings** has the entry: **group = User-Group**, the RemoteServer retrieves the **group** attribute from the **ldap** user entry for the specified user, uses the value returned, and sets the Environment variable **User-Group** to that value.
UseSSL	A boolean field indicating whether you want Cisco Access Registrar to use SSL (Secure Socket Layer) when communicating with this RemoteServer. When you set it to TRUE, be sure to specify the **CertificateDBPath** field in the **Advanced** section, and be sure the port you specified for this RemoteServer is the SSL port used by the LDAP server.

tacacs-udp

tacacs-udp specifies a TACACS server.

When you specify the **tacacs-udp** protocol, supply the information listed in Table 3-32.

Table 3-32   tacacs-udp Properties

Property	Description
MaxTries	Required. The number of times to send a proxy request to a remote server before deciding the server is off-line. You must specify a number greater than zero. The default is 3.
InitialTimeout	Required. The amount of time (in milliseconds) to wait for a response from the first proxy request. You must specify a number greater than zero. The default is 4000.

Rules

A Rule is a function that selects services based on all input information used by the function.

Advanced

Advanced objects let you configure system-level properties and the Attribute dictionary. Under normal system operation, you should not need to change the system-level properties.

> **Note** The notation *required* means Cisco Access Registrar needs a value for this property. For most of these properties, system defaults exist that you can safely use.

Table 3-33 lists the **Advanced** properties.

Table 3-33 Advanced Object Properties

Property	Description
LogServerActivity	Required. The default is FALSE, which means Cisco Access Registrar logs all responses except Access-Accepts and Access-Challenges. Accepting the default reduces the load on the server by reducing that amount of information it must log. Note, the client is probably sending accounting requests to an accounting server, so the Access-Accept requests are being indirectly logged. When you set it to TRUE, Cisco Access Registrar logs all responses to the server log file.
MaximumNumberOfRadiusPackets	Required. The default is 1024. This is a *critical property* you should set high enough to allow for the maximum number of simultaneous requests. When more requests come in than there are packets allocated, Cisco Access Registrar will drop those additional requests.
UDPPacketSize	Required. The default is 4096. RFC 2138 specifies the maximum packet length can be 4096 bytes. Do not change this value.
RequireNASsBehindProxyBeInClientList	Required. The default is FALSE. If you accept the default, Cisco Access Registrar only uses the source IP address to identify the immediate client that sent the request. Leaving it FALSE is useful when this RADIUS Server should only know about the proxy server and should treat requests as if they came from the proxy server. This may be the case with some environments that buy bulk dial service from a third party and thus do not need to, or are unable to, list all of the NASs behind the third party's proxy server. When you set it to TRUE, you must list all of the NASs behind the Proxy in the Clients list. For more information about this property, see "Using the RequireNASsBehindProxyBeInClientList Property" section on page 3-26.
AAAFileServiceSyncInterval	Required. Specified in milliseconds, the default is 75. This property governs how often the file AAA service processes accounting requests and writes the accounting records to the file. You can lower the number to reduce the delay in acknowledging the **Account-Request** at the expense of more frequent flushing of the accounting file to disk. You can raise the number to reduce the cost of flushing to disk, at the expense of increasing the delays in acknowledging the **Accounting-Request**s. The default value was determined to provide a reasonable compromise between the two alternatives.

Table 3-33   Advanced Object Properties

Property	Description
SessionBackingStoreSynchronizationInterval	Required. Specified in milliseconds, the default is 100. If you change this value it must be a number greater than zero. This property governs how often the Session Manager backing store writes updated session information to disk. You can lower the number to reduce the delay in acknowledging requests at the expense of more frequent flushing of the file containing the session data to disk. You can raise the number to reduce the cost of flushing to disk at the expense of increasing delays in acknowledging requests. The default value was determined to provide a reasonable compromise between the two alternatives.
RemoteLDAPServiceThreadTimerInterval	Required. Specified in milliseconds, the default is 10. This property governs how often the **ldap** RemoteServer thread checks to see if any results have arrived from the remote LDAP server. You can modify it to improve the throughput of the server when it proxies requests to a remote LDAP server.
MaximumNumberOfScriptingEnginesPerLanguage	Required. The default is 1. This property governs how many Tcl interpreters Cisco Access Registrar creates for processing scripts. Do not change the default value.
InitialBackgroundTimerSleepTime	Required. The default is 5. This property specifies the amount of time the time queue should initially sleep before beginning processing. This property is only used for initial synchronization and should not be changed.
MaximumNumberOfUDPTacacsPackets	Required. The default is 100. This is a critical property you should set high enough to allow for the maximum number of simultaneous proxied requests to the remote TACACS server. If more requests come in than there are packets allocated, Cisco Access Registrar will drop those additional requests.
UDPTacacsPacketsPerBlock	Required. The default is 10. This property governs how many UDP TACACS packets are allocated when Cisco Access Registrar needs to allocate more packets. You should not need to modify this value.
MinimumSocketBufferSize	Required. The default is 65536 (64 K). This property governs how deep the system's buffer size is for queueing UDP datagrams until Cisco Access Registrar can read and process them. The default is probably sufficient for most sites. You can, however, raise or lower it as necessary.
CertificateDBPath	Required if you are using an LDAP RemoteServer, and you want Cisco Access Registrar to use SSL when communicating with that LDAP RemoteServer. This property specifies the name of the file containing the client certificates to be used when establishing an SSL connection to an LDAP RemoteServer. It must be either the **cert5.db** certificate database used by Netscape Navigator 3.x (and above), or the **ServerCert.db** certificate database used by Netscape 2.x servers.

Table 3-33   Advanced Object Properties

Property	Description
LogFileSize	Required. The default is 1 Megabyte. This property specifies the maximum size of the RADIUS server log file. The value for the **LogFileSize** field is a string composed of two parts; a number, and a units indicator (<n> <units>) in which the unit is one of: K (Kilobyte, Kilobytes), M (Megabyte, Megabytes), G (Gigabyte, Gigabytes). **Note** This does not apply to the trace log.
LogFileCount	Required. The default is 2. This property specifies the number of log files to be kept on the system. A new log file is created when the log file size reaches **LogFileSize**.
UseAdvancedDuplicateDetection	Required. The default is FALSE. Set this property to TRUE when you want Cisco Access Registrar to use a more robust duplicate request filtering algorithm. For more information on this property, see the "Advance Duplicate Detection Feature" section on page 3-27.
AdvancedDuplicateDetectionMemoryInterval	Required when the Advanced Duplicate Detection feature is enabled. This property specifies how long (in milliseconds) Cisco Access Registrar should remember a request. You must specify a number greater than zero. The default is 10,000.
DefaultReturnedSubnetSizeIfNoMatch	Optional; used with the ODAP feature and reflects the returned size of the subnet if no matched subnet is found. There are three options to select if an exactly matched subnet does not exist: Bigger, Smaller, and Exact. The default is Bigger.
Ports/	Optional; allows you to use ports other than the default, 1645 and 1646. You can use this option to configure Cisco Access Registrar to use other ports,. If you add additional ports, however, Access Registrar will use the added ports and no longer use ports 1645 and 1646. These ports can still be used by adding them to the list of ports to use. For more information, refer to "Ports" section on page 3-27.
Interfaces	Optional; refer to "Interfaces" section on page 3-27
ReplyMessages	Optional; refer to "Reply Messages" section on page 3-28.
AtributeDictionary	Optional; refer to "Attribute Dictionary" section on page 3-29.
SNMP	Optional; refer to "SNMP" section on page 3-31.

Using the RequireNASsBehindProxyBeInClientList Property

You can use the property **RequireNASsBehindProxyBeInClientList** to require NASs that send requests indirectly through a proxy to be listed in the Clients list or to allow the proxy to represent them all.

- When you want to ensure the proxy is only sending requests from NASs known to this server, set the property to TRUE, and list all of the NASs using this proxy. This increases memory usage.
- When it is impossible to know all of the NASs using this proxy or when you do not care, set the property to FALSE. Cisco Access Registrar will use the proxy's IP address to identify the origin of the request.

Advance Duplicate Detection Feature

Cisco Access Registrar automatically detects and handles duplicate requests it is currently working on. It also provides an optional, more complex mechanism to handle duplicate requests that may be received by the server after it has completed processing the original request. These duplicate requests can consume extra processing power, and, if received out of order (as RADIUS is a UDP-based protocol) may cause Session Management problems.

One solution is the Advanced Duplicate Detection feature which causes Cisco Access Registrar to *remember* requests it has seen, as well as the response sent to that request, for a configurable amount of time.

To enable this feature, perform the following:

- Set the **UseAdvancedDuplicateDetection** property in the **/Radius/Advanced** section of the configuration to **TRUE**.

- Set the **AdvancedDuplicateDetectionMemoryInterval** in the **/Radius/Advanced** section to specify how long (in milliseconds) Cisco Access Registrar should remember a request.

Note Enabling this feature causes Cisco Access Registrar to keep more of its preallocated packet buffers in use for a longer period of time. The number of preallocated buffers is controlled by the **MaximumNumberOfRadiusPackets** property in the **/Radius/Advanced** section of the configuration. This property may need to be increased (which will increase the amount of memory used by Cisco Access Registrar) when the Advanced Duplicate Detection feature is enabled.

Ports

The Ports list specifies which ports to listen to for requests. When you specify a port, Cisco Access Registrar makes no distinction between the port used to receive Access-Requests and the port used to receive Accounting-Requests. Either request can come in on either port.

Most NASs send Access-Requests to port 1645 and Accounting-Requests to 1646, however, Cisco Access Registrar does not check.

When you do not specify any ports, Cisco Access Registrar does the following:

- Reads the **/etc/services** file for the ports to use for access and accounting requests.

- Otherwise, uses the standard ports (1645 and 1646).

Interfaces

The Interfaces list specifies the interfaces on which the RADIUS server receives and sends requests. You specify an interface by its IP address.

- When you list an IP address, Cisco Access Registrar uses that interface to send and receive Access-Requests.

- When no interfaces are listed, the server performs an interface discover and uses all interfaces of the server, physical and logical (virtual).

Reply Messages

The Reply Messages list allows you to choose the reply message based on the reason the request was rejected. Each of the following properties (except **Default**) corresponds to a reason why the packet was rejected. The Reply Message properties allows you to substitute your own text string for the defined errors. After you set the property (with the **set** command) and the reason occurs, Cisco Access Registrar sends the NAS that message in the Access-Reject packet as a **Reply-Message** attribute.

You might want to substitute your own messages to prevent users from getting too much information about why their requests failed. For example, you might not want users to know the password was invalid to prevent hackers from accessing your system. In such a case, you might specify the text string "unauthorized access" for the property **UserPasswordInvalid**.

Table 3-34 lists the **Reply Message** properties.

Table 3-34   Reply Message Properties

Property	Description
Default	Optional. When you set this property, Cisco Access Registrar sends this value when the property corresponding to the reject reason is not set.
UnknownUser	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever Cisco Access Registrar cannot find the user specified by **User-Name**.
UserNotEnabled	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever the user account is disabled.
UserPasswordInvalid	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever the password in the Access-Request packet did not match the password in the database.
UnableToAcquireResource	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever one of the Resource Managers was unable to allocate the resource for this request.
ServiceUnavailable	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever a service the request needs (such as a RemoteServer) is unavailable.
InternalError	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever an internal error caused the request to be rejected.
MalformedRequest	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever a required attribute (such as **User-Name**) is missing from the request.
ConfigurationError	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever the request is rejected due to a configuration error. For example, if a script sets an environment variable to the name of an object such as **Authentication-Service**, and that object does not exist in the configuration, the reason reported is ConfigurationError.
IncomingScriptFailed	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever one of the **IncomingScripts** fails to execute.
OutgoingScriptFailed	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever one of the **OutgoingScripts** fails to execute.

Table 3-34   Reply Message Properties (continued)

Property	Description
IncomingScriptRejectedRequest	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever one of the **IncomingScripts** rejects the Access-Request.
OutgoingScriptRejectedRequest	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever one of the **OutgoingScripts** rejects the Access-Request.
TerminationAction	Optional. When you set this property, Cisco Access Registrar sends back this value in the **Reply-Message** attribute whenever Cisco Access Registrar processes the Access-Request as a Termination-Action and is being rejected as a safety precaution.

Attribute Dictionary

The Attribute dictionary allows you to specify the attributes to the RADIUS server. Cisco Access Registrar comes with the standard RADIUS attributes (as defined by the RFC 2865) as well as the attributes required to support the major NASs. For more information about the standard attributes, see Appendix C, "RADIUS Attributes."

All RADIUS requests and responses consist of one or more *attributes*, such as the user's name, the user's password, the type of service the NAS should provide to the user, or the IP address the user should use for the session.

In the request and response packets, an attribute is composed of a number (between 1-255) that specifies the type of attribute to use, a length that specifies the entire attribute length, and a value. How the value is interpreted depends on its type. When it is a username, the value is a string. When it is the NAS's IP address, the value is an IP address, and so on.

Table 3-35 lists the Attribute dictionary properties.

Table 3-35   Attribute Dictionary Properties

Property	Description
Name	Required. Must be unique in the Attribute dictionary list within the same context. Although it should be an attribute defined in the RFC, the name can be any attribute defined by your client. The NAS typically comes with a list of attributes it uses. Attributes are referenced in the Profile and by Scripts by this name. The accounting file service also uses this name when printing the attribute.
Description	Optional. Description of the attribute.
Attribute	Required. Must be a number between 1-255. It must be unique within the Attribute dictionary list.
Type	Required. Must be set to one of the types listed in Table 3-36. The type governs how the value is interpreted and printed.

Types

Types are required and must be one of the following listed in Table 3-36.

Table 3-36   Types Attributes

Property	Description
UNDEFINED	Treated as a sting of binary bytes.
UINT32	Unsigned 32-bit integer.
STRING	Character string.
IPADDR	A valid IP address in dotted-decimal format.
CHAP_PASSWORD	17-byte value representing the password.
ENUM	Enums allow you to specify the mapping between the value and the strings. Once you have established this mapping, Cisco Access Registrar then replaces the number with the appropriate string. The min/max properties represent the lowest to highest values of the enumeration.
VENDOR_SPECIFIC	Vendor Specific Attribute (VSAs) are a special class of attribute. VSAs were created to extend the standard 256 attributes to include attributes required by specific manufacturers. VSAs add new capabilities for the value field in an attribute. Rather than being a simple integer string, or IP address, the value of a VSA can be one or more subattributes whose meaning depends on the vendor's definition. The Vendors list allows you to add, delete, or modify the definitions of the vendors and the subattributes they specify.

Vendor Attributes

Table 3-37 lists the **Vendor** properties.

Table 3-37   Vendor Properties

Property	Description
Name	Required. Must be unique in the Vendors attribute list.
Description	Optional. Description of the subattribute list.
VendorID	Required. Must be a valid number and unique within the entire attribute dictionary.
Type	Required. Must be one of the following: UNDEFINED, UINT32, STRING, IPADDR, CHAP_PASSWORD, ENUM, or SUB_ATTRIBUTES.

SNMP

Table 3-38 lists the five properties of the SNMP directory.

Table 3-38   SNMP Properties

Property	Description
Enabled	Either TRUE or FALSE; default is FALSE
TracingEnabled	Either TRUE or FALSE; default is FALSE
InputQueueHighThreshold	An integer; default is 90
InputQueueLowThreshold	An integer; default is 60
MasterAgnetEnabled	Either TRUE or FALSE; default is TRUE

If Enabled and MasterAgentEnabled are both TRUE, **arservagt** will start and stop the SNMP daemon (snmpd). If either of these properties is FALSE, if the AR server is not using SNMP or if your site uses a different master agent, **arservagt** will not start your master agent.

Advanced