



## Using the aregcmd Commands

This chapter describes how to use each of the **aregcmd** commands.

The Cisco Access Registrar **aregcmd** command is a command-line based configuration tool. It allows you to set any Cisco Access Registrar configurable option, as well as, start and stop the server and check statistics.

### General Command Syntax

Cisco Access Registrar stores its configuration information in a hierarchy. Using the **aregcmd** command **cd** (change directory) you can move through this information in the same manner as you would through any hierarchical file system. Or you can supply full path names to these commands to affect another part of the hierarchy, and thus avoid explicitly using the **cd** command to change to that part of the tree.

**aregcmd** command parsing is case *insensitive*, which means you can use upper or lowercase letters to designate elements. In addition, when you reference existing elements in the configuration, you need only specify enough of the element's name to distinguish it from the other elements at that level. For example, instead of typing **cd Administrators**, you can type **cd ad** when no other element at the current level begins with **ad**.

**aregcmd** command parsing is command-line order *dependent*; that is, the arguments are interpreted based on their position on the command line. To indicate an empty string as a place holder on the command line, use either single (') or double quotes ("). In addition, when you use any arguments that contain spaces, you must quote the arguments. For example, when you use the argument, "**Local Users**," you must enclose the phrase in quotes.

The **aregcmd** command can contain a maximum of 255 characters when specifying a parameter and 511 characters for the entire command.

The **aregcmd** command syntax is:

```
aregcmd [-C <clustername>] [-N <adminname>] [-P <adminpassword>]  
[-f <scriptfile>] [-v] [-q] [-p] [-n] [<command> [<args>]]
```

- **-C**—specifies the name of the cluster to log into by default
- **-N**—specifies the name of the administrator
- **-P**—specifies the password
- **-f**—specifies a file that can contain a series of commands
- **-v**—specifies verbose mode
- **-q**—turns off verbose mode

- **-p**—specifies prefix mode
- **-n**—turns off prefix mode.

Note, the verbose (**-v**) and prefix (**-p**) modes are on by default when you run **aregcmd** interactively (not running a command from the command line or not running commands from a script file). Verbose and prefix modes are off otherwise.

When you include a command (with the appropriate arguments) on the command line, **aregcmd** runs only that one command and saves any changes.

## Configuration Objects

The Cisco Access Registrar **aregcmd** command lets you manipulate configuration objects, which define properties or the behavior of the RADIUS server, such as valid administrators and types of services. For descriptions of the those objects, see Chapter 3, “Access Registrar Server Objects.”

## aregcmd Commands

This section contains the complete list of **aregcmd** commands. You can use them on the command line or insert them into scripts. The commands are listed alphabetically.

### add

Use the **aregcmd** command **add** to create new elements in the configuration. The **add** command is context sensitive, which means the type of element added is determined by the current context, or the path specified as the first parameter. The **add** command has one required argument; the name of the element you wish to add. You can also provide other parameters, or you can supply this information after **aregcmd** has added the new element. The optional second argument is a description of the element.

The syntax is:

```
add [<path>/]<name> [...]
```

### cd

Use the **aregcmd** command **cd** to change the working context, or level in the configuration hierarchy. When you use the **cd** command without any parameters, it returns you to the root of the tree. When you use the optional path argument, you can specify a new context. To change to a higher level in the tree hierarchy, use the “..” syntax (as you would in a UNIX file system). When you change to a new context, **aregcmd** displays the contents of the new location, when you are using the command in interactive mode, or if verbose mode is on.

The syntax is:

```
cd [<path>]
```

## delete

Use the **aregcmd** command **delete** to remove an element from the configuration hierarchy. You cannot remove properties on an element; you can only remove entire elements. The **delete** command is recursive; that is, it will remove any subelements contained within an element being removed. When the element is in the current context, you need only provide the name of the element to be deleted. You can optionally provide a complete path to an element elsewhere in the configuration hierarchy.

The syntax is:

```
delete [<path>/]<name>
```

## exit

Use the **aregcmd** command **exit** to terminate your **aregcmd** session. If you have any unsaved modifications, Cisco Access Registrar asks if you want to save them before exiting. Any modifications you don't choose to save are lost.

The syntax is:

```
exit
```

## filter

Use the **aregcmd** command **filter** to display a selected view of a list. You can use the **filter** command to present only the elements of a list that have properties equal to the value you specify. You can also use the **filter** command to restore the view of the list after it has been filtered.

When using the **filter** command, you must provide a property name and a value, and you can optionally provide the path to the list. Cisco Access Registrar displays a list with only those elements that have a value equal to the specified value. When you want to filter the current context, you can omit the path argument.

The **filter** command is *sticky*, in that, after you have filtered a list, you must explicitly unfilter it before you can view the complete list again. To restore the unrestricted view of the list, use the **filter** command and specify the string **all**. To restore the list in current context, you can omit the path name.

The syntax is:

```
filter [<path>] <property> <value>
```

or

```
filter [<path>] all
```

## find

Use the **aregcmd** command **find** to locate a specific item in a list. The **find** command takes one required argument, which is a full or partial pathname. After you use the command, Cisco Access Registrar displays a page beginning with the entry that most closely matches the pathname you provided.

The syntax is:

```
find <path>
```

## help

Use the **aregcmd** command **help** (with no argument specified) to display a brief overview of the command syntax. When you specify the name of a command, Cisco Access Registrar displays help for only that command.

The syntax is:

```
help [<command>]
```

## insert

Use the **aregcmd** command **insert** to add an item anywhere in ordered list. The required parameters are the numeric index of the position in the list in which you want to insert the new item, and the item value. When the list to which you are adding is not the current context, you can specify the complete path to the position in the list by prepending the path for the list to the numeric index. After the new value has been inserted into the list, Cisco Access Registrar appropriately renumbers the list.

The syntax is:

```
insert [<path>/]<index> <value>
```

This command applies to lists of servers by index and the Resource Managers list in Session Managers.

## login

Use the **aregcmd** command **login** to connect to a cluster, which contains the RADIUS server and definition of the authorized administrators. When you do not specify the cluster, admin name, and password, **aregcmd** prompts you for them.

When you are currently logged in to a cluster, the **login** command allows you to connect to another cluster. When you have changes in the current cluster that you have not saved, **aregcmd** asks if you want to save them before logging into another cluster. Any changes you do not save are lost.

After you successfully log in, and if the server is running, Cisco Access Registrar displays the cluster server's health. Note, to log in to a cluster, the AR Server Agent for that cluster must be running.

The syntax is:

```
login [<cluster>] [<name>] [<password>]
```

## logout

Use the **aregcmd** command **logout** to log out of the current cluster. After you log out, you have to log in to make any modifications to the configuration hierarchy, or to manage the server(s). When you have any unsaved modifications, Cisco Access Registrar asks if you want to save them before logging out. Any modifications you do not choose to save are lost.

The syntax is:

```
logout
```

## ls

Use the **aregcmd** command **ls** to list the contents of a level in the configuration hierarchy. This command works much like the UNIX **ls** command. When you use it without any parameters, it lists the items in the current context. When you specify a path, it lists the elements found in that context. When you use the **-R** argument, it recursively lists all of the elements in and below the specified (or current) context.

For similar commands, refer to the **next** and **prev** commands.

The syntax is:

```
ls [-R] [<path>]
```

## next

Use the **aregcmd** **next** command to review the remaining pages produced from the **ls** command. Every time you use the **cd** command, it automatically invokes the **ls** command to display the contents of the location. When the output from the **ls** command is more than one page (a page is about 24 lines) in length, Cisco Access Registrar displays only the first page.

**Note**

---

**ls** pages only user-added objects such as Users, UserLists, and attributes.

---

The **next** command takes an optional path and count. The path specifies the context in which you wish to see the next page and the count specifies the number of lines you wish to see. When you use the **next** command without the path, Cisco Access Registrar uses the current context. When you do not specify a count, Cisco Access Registrar uses the last count value you used with the **next** or **prev** command. If you never specify a count, Cisco Access Registrar uses the default value, which is 20.

Note, the current page for a context is *sticky*. This means, for example, when you use the **next** command to view entries 20 through 30, until you use the **next** or **prev** command on the same context, you will continue to see these entries even if you use the **cd** command to change to a different context, then return to the original.

The syntax is:

```
next [<path>] [<count>]
```

## prev

Use the **aregcmd** command **prev** to page backwards through the output of the **ls** command. It behaves much like the **next** command, in that it takes an optional path identifying a context to display and a count parameter indicating how many lines to display.

The syntax is:

```
prev [<path>] [<count>]
```

## pwd

Use the **aregcmd** command **pwd** to display the absolute pathname of the current context (level in the configuration hierarchy).

The syntax is:

**pwd**

## query-sessions

Use the **aregcmd** command **query-sessions** to query the server about the currently active user sessions. You can request information about all of the active sessions or just those sessions that match the type you specify.

The syntax is:

**query-sessions** *<path>* [**all**]

or

**query-sessions** *<path>* **with-*<type>*** *<value>*

Where *<path>* is the path to the server, Session Manager, or Resource Manager to query and **with-*<type>*** is one of the following: **with-NAS**, **with-User**, **with-IP-Address**, **with-IPX-Network**, **with-USR-UPN**, **with-Key**, or **with-ID**.

## quit

Use the **aregcmd** command **quit** to terminate your **aregcmd** session. You can use it interchangeably with the **exit** command.

The syntax is:

**quit**

When you quite the **aregcmd** command, if you've made changes, the Cisco Access Registrar server asks if you want to save the changes. Any unsaved changes are lost.

## release-sessions

Use the **aregcmd** command **release-sessions** to request the server to release one or more currently active user sessions. This command might be useful, for example, in the case where you have taken a NAS off-line, however, the server believes user sessions for that NAS are still active.

The syntax is:

**release-sessions** *<path>* **all**

or

**release-sessions** *<path>* **with-*<type>*** *<value>*

Where *<path>* is the path to the server, Session Manager, or Resource Manager to query and **with-*<type>*** is one of the following: **with-NAS**, **with-User**, **with-IP-Address**, **with-IPX-Network**, **with-USR-VPN**, **with-Key**, or **with-ID**.

## reload

Use the **aregcmd** command **reload** to stop the server (when it is running), and then immediately start the server, forcing it to reread its configuration information. When you have modified the configuration hierarchy, Cisco Access Registrar asks you if you want to save your changes before restarting the server. Note, you *must* save your changes in order for the reloaded server to be able to use them.

The syntax is:

```
reload
```

## reset-stats

Use the **aregcmd** command **reset-stats** to reset all server statistics displayed with the **stats** command. The **reset-stats** command provides a way of resetting the server statistics without having to reload or restart the server.

The syntax is:

```
reset-stats
```

## save

Use the **aregcmd** command **save** to validate the changes you made and commit them to the configuration database, if no errors are found.



### Note

Using the **save** command does not automatically update the running server. To update the server, you must use the **reload** command.

The syntax is:

```
save
```

Table 2-1 lists the RADIUS server objects and the effect of Dynamic Updates upon them.

**Table 2-1 Dynamic Updates Effect on Radius Server Objects**

Object	Add	Modify or Delete
Radius	Yes	Yes
UserLists	Yes	Yes
UserGroups	Yes	Yes
Policies	Yes	Yes
Clients	Yes	Yes
Vendors	Yes	Yes
Scripts	Yes	Yes
Services	Yes	Yes
SessionManagers	Yes	No
ResourceManagers	Yes	No

**Table 2-1** Dynamic Updates Effect on Radius Server Objects (continued)

Object	Add	Modify or Delete
Profiles	Yes	Yes
Rules	Yes	Yes
Translations	Yes	Yes
TranslationGroups	Yes	Yes
RemoteServers	Yes	No
Replication	Yes	Yes
Advanced	Yes	Yes
SNMP	No	No
Ports	No	No
Interfaces	No	No

## set

Use the **aregcmd** command **set** to provide values for properties on existing configuration elements. You only need to provide the **set** command with the name of the property you wish to set (or just enough of the name to distinguish it from other properties) and the new value for that property. It also applies to the **Profiles** attribute list, the Rules attributes list, the enumeration list in the Attribute dictionary, and the **LDAPtoRadiusMappings** and **LDAPtoEnvironmentMappings** mappings.

The **set** command can also be used to order servers in a list. To specify a new position in a list for a server, use the **set** command and provide the numeric position of the server and the server's name.

The syntax is:

```
set [<path>/]<property> <value>
```

When the list is a list of servers by index, the syntax is:

```
set [<path>/]<index> <server name>
```



**Note** If the index is already in use, the old server name will be replaced by the new server name.

To remove a value from a property (make a property equal to NULL), use a pair of single or double quotes as the value, as shown below:

```
set <property> ""
```

## start

Use the **aregcmd** command **start** to enable the server to handle requests. When the configuration hierarchy has been modified, Cisco Access Registrar asks you if you want to save the changes before starting the server.

The syntax is:

**start**

## stats

Use the **aregcmd** command **stats** to provide statistical information on the specified server. You can only issue this command when the server is running.

Note that **aregcmd** supports the **PAGER** environment variable. When the **aregcmd stats** command is used and the **PAGER** environment variable is set, the **stats** command output is displayed using the program specified by the **PAGER** environment variable.

The syntax is:

**stats**

The following is an example of the statistical information provided after you issue the **stats** command:

```
RemoteServer statistics for:ServerA, 209.165.201.1, port 1645
  active = TRUE
  maxTries = 3
  RTTAverage = 438ms
  RTTDeviation = 585ms
  TimeoutPenalty = 0ms
  totalRequestsPending = 0
  totalRequestsSent = 14
  totalRequestsOutstanding = 0
  totalRequestsTimedOut = 0
  totalRequestsAcknowledged = 14
  totalResponsesDroppedForNotInCache = 0
  totalResponsesDroppedForSignatureMismatch = 0
  totalRequestsDroppedAfterMaxTries = 0
  lastRequestTime = Mon Feb 18 17:19:46 2002
  lastAcceptTime = Mon Feb 18 17:18:11 2002
```

Table 2-2 lists the statistics displayed by the stats command and the meaning of the values.

**Table 2-2 aregcmd stats Information**


stats Value	Meaning
RemoteServer statistics for:	Provides server's type, name, IP address, and port used
active	Indicates whether the server was active (not in a down state)
maxTries	Number of retry attempts to be made by the RemoteServer Object based on the RemoteServer's <i>maxTries</i> property setting
RTTAverage	Average round trip time since the last server restart
RTTDeviation	Indicates a standard deviation of the RTTAverage
TimeoutPenalty	Indicates any change made to the initial timeout default value
totalRequestsPending	Number of requests currently queued
totalRequestsSent	Number of requests sent since the last server restart
	 <p><b>Note</b> totalRequestsSent should equal the sum of totalRequestsOutstanding and totalRequestsAcknowledged.</p>

Table 2-2 aregcmd stats Information (continued)

stats Value	Meaning
totalRequestsOutstanding	Number of requests currently proxied that have not yet returned
totalRequestsTimedOut	Number of requests that have timed out since last server restart or number requests not returned from proxy server within the [configured] initial timeout interval
totalRequestsAcknowledged	Number of responses received since last server restart
totalResponsesDroppedForNotInCache	Number of responses dropped because their ID did not match the ID of any Pending requests
totalResponsesDroppedForSignatureMismatch	Number of responses dropped because their response authenticator did not decode to the correct shared secret
totalRequestsDroppedAfterMaxTries	Number of requests dropped because no response was received after retrying the configured number of times. This value is different from totalRequestsTimedOut because using the default configuration values, no response within 2000 ms bumps the TimedOut counter, but it waits 14000 ms (2000 + 4000 + 8000) to bump this counter.
lastRequestTime	Date and time of last proxy request
lastAcceptTime	Date and time of last ACCEPT response to a client

## status

Use the **aregcmd** command **status** to learn whether or not the specified server has been started. When the server is running, Cisco Access Registrar displays its health.

The syntax is:

```
status
```

## stop

Use the **aregcmd** command **stop** to cause the server to no longer accept requests.

The syntax is:

```
stop
```

## trace

Use the **aregcmd** command **trace** to set the trace level in the specified server to a new value. The trace level governs how much information is displayed about the contents of a packet. When the trace level is zero, no tracing is performed. The higher the trace level, the more information displayed. The highest trace level currently used by the CAR server is trace level 5.



### Note

Although the highest **trace** level supported by the CAR server is **trace** level 5, an extension point script might use a higher level. There is no harm in setting the **trace** to a level higher than 5.

The **trace** levels are inclusive, meaning that if you set **trace** to level 3, you will also get the information reported for **trace** levels 1 and 2. If you set trace level 4, you also get information reported for **trace** levels 1, 2, and 3.

When you do not specify a server, Cisco Access Registrar sets the **trace** level for all of the servers in the current cluster. When you do not specify a value for the **trace** level, Cisco Access Registrar displays the current value of the **trace** level. The default is 0.

The syntax for setting the **trace** level is:

```
trace [<server>] [<level>]
```

Table 2-3 lists the different **trace** levels and the information returned.

**Table 2-3 Trace Levels and Information Returned**

Trace Level	Information Returned by Trace Command
0	No trace performed
1	Reports when a packet is sent or received or when there is a change in a remote server's status.
2	Indicates the following: <ul style="list-style-type: none"> <li>• Which services and session managers are used to process a packet</li> <li>• Which client and vendor objects are used to process a packet</li> <li>• Detailed remote server information for LDAP and RADIUS, such as sending a packet and timing out</li> <li>• Details about poorly formed packets</li> <li>• Details included in trace level 1</li> </ul>
3	Indicates the following: <ul style="list-style-type: none"> <li>• Error traces in TCL scripts when referencing invalid RADIUS attributes.</li> <li>• Which scripts have been executed</li> <li>• Details about local UserList processing</li> <li>• Details included in trace levels 1 and 2</li> </ul>
4	Indicates the following: <ul style="list-style-type: none"> <li>• Information about advanced duplication detection processing</li> <li>• Details about creating, updating, and deleting sessions</li> <li>• Trace details about all scripting APIs called</li> <li>• Details included in trace levels 1, 2, and 3</li> </ul>
5	Indicates the following: <ul style="list-style-type: none"> <li>• Details about use of the policy engine including: <ul style="list-style-type: none"> <li>– Which rules were run</li> <li>– What the rules did</li> <li>– If the rule passed or failed</li> <li>– Detailed information about which policies were called</li> </ul> </li> <li>• Details included in trace levels 1, 2, 3, and 4</li> </ul>

## unset

Use the **aregcmd** command **unset** to remove items from an ordered list. Specify the numeric index of the element to remove. When the ordered list is not the current context, specify the path to the list before specifying the numeric index.

When you remove an item from the list, Cisco Access Registrar rennumbers the list.

The syntax is:

```
unset [<path>/]<index>
```

This command applies to lists of servers by index, the **Profiles** attribute list, the Rules Attributes list, the enumeration list in the Attribute dictionary, and the **LDAPtoRadiusMappings** and **LDAPtoEnvironmentMappings** mappings.

## validate

Use the **aregcmd** command **validate** to check the consistency and validity of the specified server's configuration. If Cisco Access Registrar discovers any errors, it displays them.

The syntax is:

```
validate
```

# aregcmd Command Logging

**aregcmd** now records the commands that are either entered interactively, on the command line, or executed in batch mode. The recorded commands are saved in the **aregcmd\_log** file, which resides in the **logs** directory within the Cisco Access Registrar installation directory.

For security reasons, **aregcmd** blocks out the actual password that is entered as part of the command and replaces it with *<passwd>*.

In interactive mode, **aregcmd** logs the actions that are taking place in the exit/logout dialog box. The action can be **save** or **not save** if the configuration database has been modified after the last execution of the **save** command.

In non-interactive (batch or command-line) mode, **aregcmd** replaces the empty field with a NULL string.

**aregcmd** is now installed as a **setgid** program where the group is set to **staff**. This allows a non-root user to run **aregcmd** while still being able to write to the **aregcmd\_log** log file. During the installation of the Cisco Access Registrar software, you are prompted whether you want to install **aregcmd** with **setuid/setgid** permissions. You must reply "yes" unless you only run **aregcmd** as user **root**.

The following is the format of an entry in the exit/logout dialog box when **not save** has been specified:

```
Mm/dd/yyyy HH:MM:SS aregcmd Info Configuration 0 [<clustername> <username>] ( exit )
Mm/dd/yyyy HH:MM:SS aregcmd Info Configuration 0 [<clustername> <username>] ( *** New
config is not saved! ...proceed to logout.)
```

The following is sample output of an entry in the exit/logout dialog box when **not save** has been specified:

```
09/23/1999 16:18:56 aregcmd Info Configuration 0 [localhost admin] --> quit
09/23/1999 16:19:02 aregcmd Info Configuration 0 [localhost admin] --> *** New config is
not saved! ...proceed to logout.
```

The following is the format of an entry in the exit/logout dialog box when **save** has been specified:

```
Mm/dd/yyyy HH:MM:SS aregcmd Info Configuration 0 [<clustername> <username>] ( exit )
Mm/dd/yyyy HH:MM:SS aregcmd Info Configuration 0 [<clustername> <username>] ( *** New
config saved!...proceed to logout.)
```

## aregcmd Command Line Editing

Commands entered at the **aregcmd** prompt can be edited with a subset of the standard EMACS-style keystrokes. In addition, the command history may be accessed using the arrow keys on the keyboard. Use the Up arrow to retrieve the previous command and the Down arrow to retrieve the next command. A description of the supported key strokes are shown in Table 2-4.

**Table 2-4** *aregcmd Command Line Editing Keystrokes*

Key Stroke	Description
Ctrl A	Go to the beginning of the line.
Ctrl B	Move back one character.
Ctrl D	Delete the character at the cursor.
Ctrl E	Go to the end of the line.
Ctrl F	Move forward one character.
Ctrl N	Retrieve the next line.
Ctrl P	Retrieve the previous line.

