



Using the aregcmd Commands

This chapter describes how to use each of the **aregcmd** commands. The Cisco Prime Access Registrar **aregcmd** command is a command-line based configuration tool. It allows you to set any Cisco Prime Access Registrar (Prime Access Registrar) configurable option, as well as, start and stop the server and check statistics.

This chapter contains the following sections:

- [General Command Syntax](#)
- [aregcmd Commands](#)
- [aregcmd Command Logging](#)
- [aregcmd Command Line Editing](#)
- [aregcmd Error Codes](#)

General Command Syntax

Prime 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 pathnames 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 entering **cd Administrators**, you can enter **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>] [-V]  
[-f <scriptfile>] [-l <directoryname>] [-n] [<command> [<args>]] [-p] [-q] [-v]
```

- **-C**—Specifies the name of the cluster to log into by default

- **-N**—Specifies the name of the administrator
- **-P**—Specifies the password
- **-V**—Specifies view-only mode
- **-f**—Specifies a file that can contain a series of commands
- **-I**—Specifies a directory where the Prime Access Registrar license file is stored and returns information about licensed components
- **-n**—Turns off prefix mode
- **-p**—Specifies prefix mode
- **-q**—Turns off verbose mode
- **-v**—Specifies verbose mode

**Note**

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

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

This section contains the following topics:

- [View-Only Administrator Mode](#)
- [Configuration Objects](#)
- [aregcmd Command Performance](#)

View-Only Administrator Mode

Previous releases of Prime Access Registrar provided only *super-user* administrative access. If you were able to log into **aregcmd**, you could do anything to the system, including starting and stopping the system and changing the configuration. Prime Access Registrar provides view-only administrative access. View-only access restricts an administrator to only being able to observe the system and prevents that user from making changes.

View-only access can be encountered in three ways:

- Specific administrators can be restricted to view-only access whenever they log in.
- Administrators not restricted to view-only access can choose to start **aregcmd** in a view-only mode. This might be used when an administrator wants to ensure that he or she does not make any changes.
- When an administrator who is not view-only logs in to a slave server, they will be unable to make changes to any parts of the configuration other than **/Radius/Replication**, **/Radius/Advanced/Ports**, **/Radius/Advanced/Interfaces** or the properties in **/Radius/Advanced**. This is because the rest of the configuration is replicated from the master server and changes directly to the slave will cause problems.

**Note**

When a user logs in, the system determines whether a user's session is view-only or not. If the configuration is changed after a user has logged in, that change does not take effect until the affected user logs out and logs back in.

ViewOnly Property

The ViewOnly property has been added to the Administrators configuration. The default setting for the ViewOnly property is FALSE. The following shows the default setting for the **admin** user:

```
cd /Administrators/admin
```

```
[ //localhost/Administrators/admin ]
Name = admin
Description =
Password = <encrypted>
ViewOnly = FALSE
```

You can designate specific administrators to be view-only administrators by setting the new ViewOnly property to TRUE.

- If that property is set to TRUE, any time the administrator logs in to **aregcmd** the session will be in view-only mode.
- If set to FALSE, when the administrator logs in to a master server, the session will be full super-user capability.

If the administrator logs in to a slave, they only part of the configuration they will be able to modify is that part under **/Radius/Replication**, **/Radius/Advanced/Ports**, **/Radius/Advanced/Interfaces** or the properties in **/Radius/Advanced**.

When in a view-only session, the following commands will cause an error: **add**, **delete**, **set**, **unset**, **insert**, **validate**, **save**, **start**, **stop**, **reload**, **reset-stats**, **release-sessions**, and **trace**. The following error message will be displayed:

```
316 Command failed: session is ViewOnly
```

When in a slave server session, the following commands will cause an error when the object or property being operated on is not under **/Radius/Replication**, **/Radius/Advanced/Ports**, **/Radius/Advanced/Interfaces** or the properties in **/Radius/Advanced**: **add**, **delete**, **set**, **unset**, and **insert**. The following error message will be displayed:

```
317 Command failed: session is ViewOnly
```

Configuration Objects

The Prime Access Registrar **aregcmd** command lets you manipulate configuration objects, that define properties or the behavior of the RADIUS server, such as valid administrators and types of services. For descriptions of those objects, see [Chapter 4, “Cisco Prime Access Registrar Server Objects.”](#)

aregcmd Command Performance

You can impact **aregcmd** command performance and server response time by having Prime Access Registrar userlists that contain more than 10,000 users. Prime Access Registrar userlists were not designed to contain 10,000 users in any one list.

If you must provide service for groups greater than 10000 users, we recommend that you use an external data store such as an **LDAP directory** or an **Oracle database**. If you are unable to use an external data store, create multiple userlists instead, keeping each userlist under 10,000 users.

Multiple userlists require multiple services (one for each userlist), because a service cannot reference more than one userlist. The multiple services can then be combined using the Service Grouping feature with ResultRule, OR, as follows:

```
[ //localhost/Radius/Services/GroupService ]
  Name = GroupService
  Description =
  Type = group
  IncomingScript~ =
  OutgoingScript~ =
  ResultRule = OR
  GroupServices/
  1. UserService1
  2. UserService2
  3. UserService3
```

RPC Bind Services

The Prime Access Registrar server and the **aregcmd** CLI requires RPC services to be running before the server is started. If the RPC services are stopped, you must restart RPC services, then restart the Prime Access Registrar server.

Use the following commands to restart RPC services:

```
arserver stop
```

```
/etc/init.d/rpc start
```

```
arserver start
```

If RPC services are not running, the following message is displayed when you attempt to start aregcmd:

```
Login to aregcmd fails with the message:
400 Login failed
```

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.

This section contains the following topics:

- [add](#)
- [cd](#)
- [delete](#)
- [exit](#)
- [filter](#)
- [find](#)
- [help](#)
- [insert](#)
- [login](#)
- [logout](#)

- `ls`
- `next`
- `prev`
- `pwd`
- `query-sessions`
- `quit`
- `release-sessions`
- `reload`
- `reset-stats`
- `save`
- `set`
- `start`
- `stats`
- `status`
- `stop`
- `tacacs-stats`
- `tacacs-reset-stats`
- `dia-stats`
- `trace`
- `trace-file-count`
- `unset`
- `validate`

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, Prime 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. Prime 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 pathname.

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, Prime 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, Prime 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, Prime 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, Prime Access Registrar displays the cluster server's health. Note, to log into a cluster, the Prime Access Registrar 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 into make any modifications to the configuration hierarchy, or to manage the server(s). When you have any unsaved modifications, Prime 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, Prime Access Registrar displays only the first page.



Note

ls command retrieves 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, Prime Access Registrar uses the current context. When you do not specify a count, Prime Access Registrar uses the last count value you used with the **next** or **prev** command. If you never specify a count, Prime 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> [send-CoA [with-profile <profile name>] ]
```

or

```
query-sessions <path> with-Attribute <name> <value> [send-CoA [with-profile <profile name>] ]
```

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**, **with-ID** or **with-Age**. The optional **[with-profile <profile name>]** parameter indicates a profile name as configured in **/Radius/Profiles**.

The **query-sessions** command with an optional **[send-CoA]** at the end causes the Prime Access Registrar server to send a Change of Authorization (CoA) request to the client. The CoA request includes the CoA attributes configured for the client. When the optional profile name is also included in the command, the Prime Access Registrar server includes the attribute-value (AV) pairs from the corresponding profile in **/Radius/Profiles** in the CoA request.

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 quit the **aregcmd** command, if you have made changes, the Prime 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 offline, however, the server believes user sessions for that NAS are still active.

The syntax is one of:

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

or

release-sessions *<path>* **with-** *<type>* *<value>* [**send-pod**] [**send-notification**]

or

release-sessions *<path>* **with-Attribute** *<name>* *<value>* [**send-pod**] [**send-notification**]

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**.

The optional [**send-pod** **<send notification>**] parameter sends the disconnect packet to the NAS to clear sessions and an Accounting-Stop notification to the client listed in the session record.

The optional **with-Attribute** parameter enables release a session based on a specific attribute and value.

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, Prime Access Registrar asks you if you want to save your changes before restarting the server. 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 also resets SNMP counters.

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 |
| 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> ""
```

When you need to set an attribute to a value that includes a space, you must double-quote the value, as in the following:

```
set Framed-Route "192.168.1.0/24 192.168.1.1"
```

start

Use the **aregcmd** command **start** to enable the server to handle requests. When the configuration hierarchy has been modified, Prime 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 2013
  lastAcceptTime = Mon Feb 18 17:18:11 2013
```

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 Note totalRequestsSent should equal the sum of totalRequestsOutstanding and totalRequestsAcknowledged. |
| 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. |

Table 2-2 *aregcmd stats Information (continued)*

| Stats Value | Meaning |
|-----------------|---|
| 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, Prime 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
```

tacacs-stats

Use the **aregcmd** command **tacacs-stats** to provide statistical information of TACACS+.

The syntax is:

```
tacacs-stats
```

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

```
Global Tacacs+ Statistics
  serverStartTime = Mon Apr 15 01:17:34 2013
  serverResetTime = Mon Apr 15 01:17:34 2013
  serverState = Running
  totalPacketsReceived = 60
  totalPacketsSent = 60
  totalRequests = 60
  totalResponses = 60
  totalAuthenticationRequests = 2
  totalAuthenticationAccepts = 2
  totalAuthenticationRejects = 0
  totalAuthenticationChallengeRequests = 0
  totalAuthenticationResponses = 2
  totalAuthorizationRequests = 56
  totalAuthorizationAccepts = 38
  totalAuthorizationRejects = 18
  totalAuthorizationResponses = 56
  totalAccountingRequests = 2
  totalAccountingAccepts = 2
  totalAccountingRejects = 0
  totalAccountingResponses = 2
  totalPacketsInUse = 0
```

```
totalPacketsDropped = 0
```

See [TACACSStatistics](#) for more information.

tacacs-reset-stats

Use the **aregcmd** command **tacacs-reset-stats** to reset TACACS+ statistics displayed with the **stats** command. The **tacacs-reset-stats** command also resets SNMP counters.

The **tacacs-reset-stats** command provides a way of resetting the TACACS+ statistics without having to reload or restart the server.

The syntax is:

```
tacacs-reset-stats
```

dia-stats

Use the **aregcmd** command **dia-stats** to provide statistical information of Diameter.

The syntax is:

```
dia-stats
```

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

```
dia-stats /Radius/RemoteServers/dia
```

```
Diameter Remote server statistics for: dia, 10.81.79.76, port 3868
active = TRUE
cDiaRemSvrRTTAverage = 25ms
cDiaRemSvrRTTDeviation = 0ms
cDiaRemSvrServerType = Diameter
cDiaRemSvrTotalRequestsPending = 0
cDiaRemSvrTotalRequestsOutstanding = 0
cDiaRemSvrStatsState = Closed
cDiaRemSvrStatsASRsIn= 0
cDiaRemSvrStatsASRsOut= 0
cDiaRemSvrStatsASAsIn = 0
cDiaRemSvrStatsASAsOut = 0
cDiaRemSvrStatsACRsIn = 0
cDiaRemSvrStatsACRsOut = 0
cDiaRemSvrStatsACAsIn = 0
cDiaRemSvrStatsACAsOut = 0
cDiaRemSvrStatsCERsIn = 0
cDiaRemSvrStatsCERsOut = 0
cDiaRemSvrStatsCEAsIn = 0
cDiaRemSvrStatsCEAsOut = 0
cDiaRemSvrStatsDWRsIn = 0
cDiaRemSvrStatsDWRsOut = 0
cDiaRemSvrStatsDWAsIn = 0
cDiaRemSvrStatsDWAsOut = 0
cDiaRemSvrStatsDPRsIn = 0
cDiaRemSvrStatsDPRsOut = 0
cDiaRemSvrStatsDPAsIn = 0
cDiaRemSvrStatsDPAsOut = 0
cDiaRemSvrStatsRARsIn = 0
cDiaRemSvrStatsRARsOut = 0
cDiaRemSvrStatsRAAsIn = 0
cDiaRemSvrStatsRAAsOut = 0
```

```

cDiaRemSvrStatsSTRsIn= 0
cDiaRemSvrStatsSTRsOut = 0
cDiaRemSvrStatsSTAsIn = 0
cDiaRemSvrStatsSTAsOut = 0
cDiaRemSvrStatsRedirectEvents = 0
cDiaRemSvrStatsAccDupRequests = 0
cDiaRemSvrStatsMalformedRequests = 0
cDiaRemSvrStatsAccsNotRecorded = 0
cDiaRemSvrStatsWhoInitDisconnect = 0
cDiaRemSvrStatsAccRetrans = 0
cDiaRemSvrStatsTotalRetrans= 0
cDiaRemSvrStatsAccPendRequestsOut = 0
cDiaRemSvrStatsAccReqstsDropped = 0
cDiaRemSvrStatsHByHDropMessages = 0
cDiaRemSvrStatsEToEDupMessages= 0
cDiaRemSvrStatsUnknownTypes= 0
cDiaRemSvrStatsProtocolErrors = 0
cDiaRemSvrStatsTransientFailures = 0
cDiaRemSvrStatsPermanentFailures = 0
cDiaRemSvrStatsDWCurentStatus= 0
cDiaRemSvrStatsTransportDown = 0
cDiaRemSvrStatsTimeoutConnAtmpts = 5

```

dia-stats-reset

Use the **aregcmd** command **dia-stats-reset** to reset Diameter statistics displayed with the **stats** command. The **dia-stats-reset** command also resets SNMP counters.

The **dia-stats-reset** command provides a way of resetting the Diameter statistics without having to reload or restart the server.

The syntax is:

```
dia-stats-reset
```

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 Prime Access Registrar server is trace level 5.



Note

Although the highest **trace** level supported by the Prime Access Registrar 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. However, increasing the trace level impacts the system performance.

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, Prime 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, Prime 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"> • Which services and session managers are used to process a packet • Which client and vendor objects are used to process a packet • Detailed remote server information for LDAP and RADIUS, such as sending a packet and timing out • Details about poorly formed packets • Details included in trace level 1 |
| 3 | Indicates the following: <ul style="list-style-type: none"> • Error traces in TCL scripts when referencing invalid RADIUS attributes. • Which scripts have been executed • Details about local UserList processing • Details included in trace levels 1 and 2 |
| 4 | Indicates the following: <ul style="list-style-type: none"> • Information about advanced duplication detection processing • Details about creating, updating, and deleting sessions • Trace details about all scripting APIs called • Details included in trace levels 1, 2, and 3 |
| 5 | Indicates the following: <ul style="list-style-type: none"> • Details about use of the policy engine including: <ul style="list-style-type: none"> – Which rules were run – What the rules did – If the rule passed or failed – Detailed information about which policies were called • Details included in trace levels 1, 2, 3, and 4 |

trace-file-count

Use the **aregcmd** command **trace-file-count** to change the trace log file count dynamically without requiring a server reload. The syntax is:

trace-file-count *n*

Where *n* is a number that specifies the number of trace log files. This function is helpful for debugging situations when you do not want to perform a **reload**.

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, Prime 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 Prime Access Registrar discovers any errors, it displays them.

The syntax is:

```
validate
```

OpenSSL Commands

This section contains a list of **OpenSSL** commands. You can use them on the command line or insert them into scripts.

This section contains the following topics:

- [ecparam](#)
- [req](#)
- [ca](#)

ecparam

Use the **OpenSSL** command **ecparam** to manipulate or generate elliptical curve (EC) parameter files.

The syntax is:

```
ecparam
```

req

Use the **OpenSSL** command **req** to create and process certificate requests.

The syntax is:

```
req
```

ca

Use the **OpenSSL** command **ca** used to sign certificate requests and generate CRLs it also maintains a text database of issued certificates and their status.

The syntax is:

```
ca
```

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 Prime 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 Prime 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:

```
10/12/2013 16:18:56 aregcmd Info Configuration 0 [localhost admin] --> quit
10/12/2013 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 can 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. |

aregcmd Error Codes

[Table 2-5](#) lists the error codes used in **aregcmd**.

Table 2-5 *aregcmd Error Codes*

| Error Code | Meaning |
|------------|--|
| 200 | OK |
| 300 | Command failed to parse; usually an unbalanced quotation |
| 301 | Unknown command; usually caused by a misspelled command |
| 302 | Ambiguous command; characters you have entered select more than one command |
| 303 | Not logged in; you have logged out of aregcmd and attempted another aregcmd command |
| 304 | Too few arguments |
| 305 | Too many arguments |
| 306 | Invalid argument |
| 307 | Object not found or path ambiguous; you have tried to set an unknown object or provided a partial name that matches multiple objects |
| 308 | Object already exists |
| 309 | Confirmation password did not match; when setting a user password, the re-entered password did not match the initial entry |
| 310 | Command failed; a generic response for a command that failed for a reason other than those listed here |
| 311 | Command is interactive; possibly due to attempting to batch mode with an interactive command |
| 312 | Validation failed; a new or modified object is not valid |

Table 2-5 aregcmd Error Codes (continued)

| Error Code | Meaning |
|------------|--|
| 313 | Failed to reread; an error occurred while synchronizing changes from another aregcmd session; occurs only when using multiple aregcmd instances |
| 314 | Failed to open the pager; the PAGER environment variable is set to something that does not exist and the stats command is entered |
| 315 | Property is not modifiable; an administrator has attempted to modify a read-only property |
| 316 | Command failed: session is ViewOnly; an view-only administrator has attempted to modify a property |
| 317 | Command failed: server is a Replication Slave; an administrator has attempted to modify a replicated property on a slave of an SMDBR network |
| 400 | Login failed; an incorrect username or password was given during aregcmd log in |
| 401 | Unable to access server; aregcmd was unable to communicate with the radius process usually because the process is not running or is otherwise unresponsive |
| 402 | Login failed: version of aregcmd is incompatible with server; a new version of aregcmd has tried to connect with an older version of Prime Access Registrar |
| 500 | Internal error; a generic aregcmd error |

