RSA® Authentication Manager 8.4 Bulk Administration Utility Guide

Revision 5

Contact Information

RSA Link at **https://community.rsa.com** contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

Dell, RSA the RSA Logo, EMC and other trademarks, are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners. For a list of RSA trademarks go to w**ww.emc.com/legal/emc-corporation-trademarks.htm#rsa**.

License Agreement

This software and the associated documentation are proprietary and confidential to Dell Inc. or its subsidiaries are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability.

This software is subject to change without notice and should not be construed as a commitment by Dell, Inc.

Third-Party Licenses

This product may include software developed by parties other than RSA. The text of the license agreements applicable to third-party software in this product may be viewed on the product documentation page on RSA Link. By using this product, a user of this product agrees to be fully bound by terms of the license agreements.

Note on Encryption Technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Use, copying, and distribution of any Dell software described in this publication requires an applicable software license.

Dell Inc. believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS". DELL INC. MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

December 2018
Revised: May 2020

Contents

Revision History

Revision Number	Date	Revision
1	February 2019	Patch 1 added new functionality that was first introduced in version 8.3 Patch 3 and Patch 4: • The **SoftTokenProfile** attribute and the **SSTP** (Set Software Token Profile) command for software token distribution with CT-KIP and SDTID files. • The **AgentNewHostName** attribute for the **AAH** (Add Agent Host) command. • The **ETL** (Extend Software Token Lifetime) command extends the lifetime of software tokens. The TokSerial field is required. • The **CPADC** (Change Principal Attribute Data Clear) command clears existing custom attribute data for custom attributes that are assigned to a principal (user). Removed the descriptions of token attributes that are no longer used with RSA Authentication Manager 8.2 or later: **OTPLength**, **OTPAlgorithm**, **OTPinterval**, and **PinAdded**.
2	April 2019	Patch 3 added an optional **ChangePwdFlag** field that allows uo to force a password change while using the **AU** action to add a User.
3	September 2019	Patch 6 added new commands to update the RSA Authentication Manager internal database: • The **ARG** (Add Remote Group) command adds user groups from a trusted realm. • The **DRG** (Delete Remote Group) command removes a remote user group. • The **AURG** (Add User to Remote Group) command links a trusted user to a trusted user group. • The **DURG** (Delete User from Remote Group) command removes a trusted user from a trusted user group. • The **ARGC** (Add Remote Group Client) command links an authentication agent to a trusted group. • The **DRGC** (Delete Remote Group Client) command removes an authentication agent from a trusted user group Patch 6 added an optional **UseSameSecurityDomain** field to the **ETL** (Extend Software Token Lifetime) command that requires AMBA to only select unassigned software tokens that share the same security domain as the tokens that are being extended.

Revision Number	Date	Revision
4	March 2020	Patch 11 added a new **ARC** (Add RADIUS Client) command for adding IPv4 RADIUS clients, and added security domains to the results file for the **LTIF** (List Token Information by Field) command and the **LUIF** (List User Information by Field) command.
5	June 2020	Patch 13 updates the **CAU** (Change or Add User) command with a new RemoveGrpDefLogin option to remove a logon alias from a group.

Preface

About This Guide

This document describes how RSA® Authentication Manager 8.4 (Authentication Manager) administrators can use the Authentication Manager Bulk Administration (AMBA) utility developed from the Authentication Manager Server Admin APIs. This utility enables administrators to perform administration from the command-line.

For a complete list of documentation, see "RSA SecurID Access Product Documentation" on RSA Link at **https://community.rsa.com/docs/DOC-60094**.

For a description of common Authentication Manager terms, see the "RSA Authentication Manager Glossary" on RSA Link at **https://community.rsa.com/docs/DOC-76682**.

Support and Service

You can access community and support information on RSA Link at **https://community.rsa.com**. RSA Link contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Support for RSA Authentication Manager

Before you call RSA Customer Support for help with the Authentication Manager Appliance, have the following information available:

- Access to the Authentication Manager Appliance.
- Your license serial number. To find this number, do one of the following:
  - Look at the order confirmation e-mail that you received when your ordered the product. This e-mail contains the license serial number.
  - Log on to the Security Console, and click **License Status**. Click **View Installed License**.
- The Authentication Manager Appliance software version. To find this version number, do one of the following:
  - Look in the top, right corner of the Quick Setup,
  - Log on to the Security Console and click **Software Version Information**.

Support for the Cloud Authentication Server and Identity Routers

If your company has deployed identity routers and uses the Cloud Authentication Service, RSA provides you with a unique identifier, called the Customer Support ID, which is required when you register with RSA Customer Support.
To see your Customer Support ID, sign in to the Administration Console and click **My Account > Company Settings**.

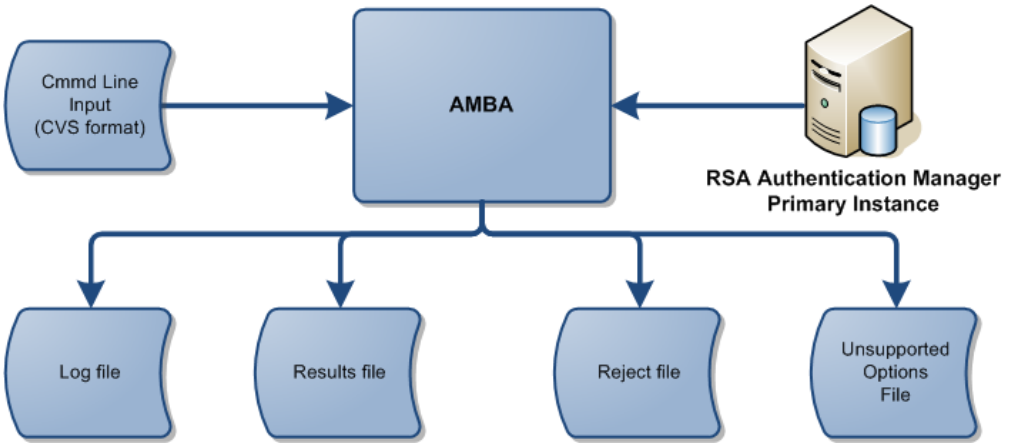
RSA Ready Partner Program

The RSA Ready Partner Program website at **https://www.rsa.com/en-us/partner/technology-partners** provides information about third-party hardware and software products that have been certified to work with RSA products. The website includes Implementation Guides with step-by-step instructions and other information on how RSA products work with third-party products.

1 Overview of the AMBA Utility

The AMBA utility supplements administrative features of RSA Authentication Manager (Authentication Manager) and was developed from Authentication Manager Server Admin APIs. The utility enables Authentication Manager administrators to perform bulk administration functions from the command-line interface.

The following diagram provides an overview of the AMBA utility.



The AMBA utility implements a sub-set of common functions available through the Authentication Manager administrative consoles.

Important: Protect this application with the appropriate operating system access rights. Use of this utility by unauthorized persons could lead to loss of data and denial of service to affected Users.

The topics in this chapter include:

Execute the AMBA Utility Command-line Options

Use the **rsautil** command to execute the AMBA utility command-line options.

Before You Begin:

- The AMBA utility requires a valid Enterprise Edition or Premium Edition license file, unless you have upgraded to the current release of Authentication Manager with a Base Server license that includes the AMBA utility add-on option.
- Verify that Authentication Manager is running. For example, log on to the Security Console.
- Ensure you have Super Admin access.
- Obtain the **rsaadmin** operating system password for the primary instance.
- Enable Secure shell (SSH) for Authentication Manager:
  - Log on to the Operations Console as an Operations Console administrator, and click **Administration > Operating System Access**.
  - Click on **Enable SSH Access**, if unchecked, for one or both interfaces and click **Save**.

Note: Authentication Manager uses a Linux operating system. All commands are case sensitive.

Procedure:

1. Launch the SSH client.
2. Connect to the primary instance using the IP address or fully qualified hostname.
3. Enter the operating system User ID, **rsaadmin**.
4. At the prompt, enter the password for the **rsaadmin** account.
5. Change directories to **/opt/rsa/am/utils**:
   `cd /opt/rsa/am/utils`
6. Run the AMBA utility:
   `./rsautil AMBulkAdmin options`
   where *options* are Command-line Options.
7. At the prompt, enter the Super Admin User ID.
8. At the prompt, enter the Super Admin password.
9. To exit the AMBA utility, close the SSH client:
   `exit`

Usage

The AMBA utility can be used with the following supported standalone options:

```
rsautil AMBulkAdmin
rsautil AMBulkAdmin -v
rsautil AMBulkAdmin --gtc templatename
rsautil AMBulkAdmin --ini inifile
rsautil AMBulkAdmin null
```

Note: These options are mutually exclusive of any other option. If one of these options is specified, any additional options are ignored.

The following supported options can be used together:

```
rsautil AMBulkAdmin [-a SAUserId] [--debug] [-g | --ctkip]
[--gdir directory][-i datafile] [--lic license file][-m value]
[--newlog] [--nolog] [-o results file] [-p value] [-P value]
[-r results file] [--rej command reject file] [--verbose]
[--userPwd user password] [-x value]
```

[…] denotes an optional parameter, not to be included in the actual data.

| denotes a choice.

Command-line Options

The order of the options is not critical. One or more spaces are required between the option key and its value.

CAUTION: Command-line variables containing spaces must be enclosed in double quotes. For example, “E MMM dd HH:dd:ss z yyyy.”
The bash shell on the appliance interprets the ‘**!**’ character as a special character referencing the shell’s command buffer. Other UNIX/LINUX shells can have the same or different side effects. One work-around for this is to use the INI switch and put these parameters into an .ini file. For more information, see Input Parameter File.

null

Displays an input options usage report on **stderr**.

-a *SuperAdminUserID*

where *SuperAdminUserID* is the Super Admin User Id.

Provides the Super Admin User Id.

--ctkip

Enables the generation of CT-KIP credentials for tokens assigned during **AMBulkAdmin** processing. A CT-KIP activation code and download URL are generated.

This option and the **-g** option are mutually exclusive.

--datefmt *DateFormat*

Overrides the formatting applied to List elements displaying a Java Date object. The default format is E MMM dd HH:mm:ss z yyyy which displays a date in the form “Mon Mar 07 23:18:03 EST 2011.” For more information, see Change Date Format.

--debug

Disables all calls to the Authentication Manager Server API and forces a successful return result. This option allows:

- An input file to be processed without making any changes to the database, or various features to be tested when a database is not present.
- An input file to be validated for required fields without making changes to the database. The option does not perform any of the validations performed by Authentication Manager, such as rejecting an attempt to add a User that already exists in the database.

-g

Turns on the option to output RSA SecurID (SecurID) software token database files for tokens assigned during **AMBulkAdmin** processing. File names are based on the default login name and token serial number of the User, and have the extension .sdtid. For example, **juser_000027050105.stdid** where the default login name is *juser* and the assigned token serial number is *27050105*.

By default, files are stored in the current directory. Use the **-gdir** option to change the directory.

The **-g** and **--ctkip** options are mutually exclusive.

--gdir *dirname*

where *dirname* is either an absolute path **/opt/rsa/am/amba/sdtidfiles**, or the relative path from the directory where the AMBA utility is run. Must be fewer than 128 characters.

Specifies that SecurID software token database files are stored in the specified directory. The **-g** option must be specified to cause the database files to be saved. If the **-g** command-line option is not used, any **-gdir** command-line option is ignored.

--gtc *templatename*

where *templatename* is a fully qualified path and file name. Must be fewer than 128 characters.

Generates a CSV template file to develop a data file. The generated file contains a header line, a line with the correct number of empty columns, and one or more comment lines. For more details, see Input File Templates.

-i datafile | stdin

where *datafile* is a fully qualified path. Must be fewer than 128 characters.

Defines the path to the CSV formatted input file. For more details, see Data File. The literal **stdin** redirects system standard input into the AMBA utility.

--ini *inifile*

where *inifile* is the path and filename of the input parameter file. Must be fewer than 128 characters.

The input parameter file is a text file containing input options and parameters. For more details about this file, see Input Parameter File. Additional command-line options have precedence over duplicate input parameter file options.

--lic *license file*

where *license file* is the name of the license file to be used. The path is relative to the current working directory from which the AMBA utility is run, unless an absolute path is provided.

Provides the location of the AMBA utility license file, **AMBAlicense.dat**. If not specified, the file must be in the current AMBA utility directory.
This is necessary only for Base license users.

-m 0 | 1 | 2 | 3

Sets the log messaging level. The following table summarizes the log level actions for the message types:

Message Type	Level			
	0	1	2	3
Boj	Yes	Yes	Yes	Yes
Eoj	Yes	Yes	Yes	Yes
Error	Yes	Yes	Yes	Yes
Failure	Yes	No	Yes	Yes
Info	Yes	No	No	No
Success	Yes	Yes	No	Yes

For more information about logging and an example of a log with each message type, see Log File.

--newlog

Forces the AMBA utility to create a new log on each execution of the AMBA utility, so that any existing log information is overwritten.

--nolog

Turns off all AMBA logging. Authentication Manager logging is not affected.

-o [*log file* | stderr | stdout]

where *log file* is a fully qualified path and filename. Must be fewer than 128 characters.

Defines the destination of the log output. The default is **AMBulkAdminlog.txt**, created in the current directory. The literals **stderr** and **stdout** redirect output to system standard error or standard output files.

By default, new log information is appended to any previous log information. For more information, see Log File.

-p 1 | 2 | . . . 3600

Enables the display of a progress report and the time delay in seconds between updates. If enabled, the progress report displays on **STDERR**.

-P *SuperAdminPassword*

where *SuperAdminPassword* is the Super Admin password.

Provides the Super Admin password.

-r [*results file* | stdout | stderr]

where *results file* is the path and file name.

Defines the destination for the results of a **List** action. The results file is overwritten on each execution of the AMBA utility. The default is **AMBulkAdminResults.CSV,** created in the current directory. The literals **stderr** and **stdout** redirect output to system standard error or standard output files. For more details, see Results File.

--rej [*command reject file]*

where *command reject file* is a fully qualified path and file name. Must be fewer than 128 characters.

Specifies the path and file name of the file which contains rejected input records, in the same format as the input file. The default is **AMBulkAdminReject,** created in the current directory. If the default file name is used, the file extension of the input file is appended to it. For more details, see Command Reject File.

--searchlimit

Determines the maximum number of principal objects returned by the **SearchPrincipals** command. The default value is 20,000. If the number of principals in the database is greater than 20,000, set the search limit to something slightly larger (+100) than the number of principals in the database. Failure to do so truncates the results of various commands.

--userPwd *user password*

where *user password* is a password value that meets the defined password policy requirement.

When adding or changing multiple Users, this option gives all the applicable Users the same password. Without this option, a password for each User is provided in the input file in the **UserPwd** field.

Important: If the input file contains multiple actions that require a User password, then the *user password* is used for all those actions. Verify the input file data before using this option.

-v

Returns the AMBAutility version number. The **-version** option is not supported.

--verbose

Enables enhanced logging. This function is usually only for debugging. It generates Information message types and records program flow.

Note: Use of this option can severely degrade program performance.

-x 0 | 1

Enables/disables the insertion of a datestamp into the name of the log, command reject, results and software token database files. The default is 0.

Command-line Options Summary

The following table summarizes the supported command-line options:

Table 10: Supported Command-line Options

Option	Description
null	Displays input options and its usage.
-a	Sets the name of Super Admin User Id.
--ctkip	Enables the generation of CT-KIP credentials for tokens.
--datefmt	Sets the format applied to List elements displaying a Java Date object.
--debug	Allows input file validation for required fields without making changes to the database.
-g	Saves the SecurID software tokens to database files.
--gdir *dirname*	To store the SDTID files in the specified director.
--gtc *templatename*	Creates a template file in CSV format.
-i *datafile* | stdin	Provides the input data file.
--ini *inifile*	Sets the path to the input parameter file.
--lic *license file*	Sets the license file location.
-m 0 | 1 | 2 | 3	Sets the message logging level.
--newlog	Forces the creation of a new log file.
--nolog	Turns off all AMBA logging.
-o [*log file* | stderr | stdout]	Specifies the path to a file for storing the log output.
-p 1 | 2 | . . . 3600	Enables the display of a progress report with a time delay.
-P *value*	Password for specified admin User.
-r [*results file* | stdout | stderr]	Defines file for storing the results of a **List** actions.
--reg [*command reject file*]	Specifies the file which contains rejected input records.
--searchlimit *value*	Sets the maximum number of principal objects returned by the **SearchPrincipals** command.
-v	Displays the AMBA utility version number.
--verbose	Enables enhanced logging.
--userPwd	Assigns a password to create an IMS user.
-x 0 | 1	Enables/disables insertion of a datestamp into the file names.

Return Values

The AMBA utility returns **0** if no errors are detected, or a number greater than **0** if **Error** or **Failure** messages are detected and reported.

The return value has the following meaning:

0 = no errors

1 = one or more Error messages

2 = one or more Failure messages

3 = one or more Error and Failure messages

4 = System error, message is sent to **STDOUT**.

An **Error** message in the output log is usually the result of a sub-command failure. This often results in a command failure. There can be multiple **Error** messages for a single command. For one or more **Error** messages, a value of **1** is returned.

A command failure produces a **Failure** message in the output log. For one or more **Failure** messages, a value of **2** is returned.

If both **Error** and **Failure** messages have been detected, then a value of **3** is returned.

Note: The return values are intercepted by **rsautil**. To return these values to a calling task such as a script, use either the **-S** or **--script-exit** option. For example:

```
rsautil --script-exit AMBulkAdmin parameter list
```

This causes **rsautil** to pass any AMBA return values back to the calling process.

For more details about logging, see Log File.

2 File Processing

This chapter introduces the files associated with the AMBA utility, and provides details of how these files are used.

Data File

The data file consists of an optional header line and a required data line for every action to be executed by AMBA.

- The header line is a template for the data line. It indicates what data is present and the order in which the data appears in the data line. The fields used are dependent on the action to be executed. If present, the header line must be the first line of input and the first field must be **Action** or **"Action"**. The header line is not case sensitive.
- The data line consists of an action code, required fields and zero or more optional fields. For an action to succeed, data for all required fields must be present in the data line. Some optional fields execute a subordinate action.

For more information about the supported action codes, see Action Codes.

The input fields fall into two categories:

- **Required** - The action cannot be performed without this data and returns an error if the data is missing.
- **Optional -** The action uses the data, if provided.

For a list and definitions of the available input fields, see Input Field Definitions.

Prepare the Data File

Data files are created in CSV format, in a spreadsheet or word processor application that is capable of saving in that format, such as Microsoft Excel.

Do not to leave the first line in the data file empty or blank. The utility ignores leading blank lines. In the log information, the input lines are numbered from the first non-blank line, which is treated as line 1.

```
Action,LastName,FirstName,DefLogin,GrpName
<action code>,<lastname data>,<firstname data>,<grpname data>
```

In this example, the first non-blank line is the header line, made up of input field names. The second non-blank line is a data line, made up of input data corresponding to the header line input fields.

Where a header line is included, it determines which input fields are present for the specified action.

Where a header line is **not** included, the AMBA utility assumes that all the input fields for the specified action are present in every line, with "action" as the first field.

Each input field is separated by a comma, with no double quotes (").

The data line can contain empty optional input fields. The empty fields are delimited by ,, (comma comma). A data file without a header must specify all empty fields. In a data file with a header line, is not necessary for the data lines to specify empty fields beyond the last input field that contains data. For an example, see Sample Data File.

Comments can be placed anywhere in the input file. A comment is any line beginning with two forward slashes "//".

Action Sequence

The data file is processed in sequential order. The RSA Authentication Manager administrator must arrange transactions in a logical order to ensure any subordinate actions succeed. For example, if a User is being added and the Group field contains an entry, then the Group must already exist for the association to be successful.

- If the primary action fails, the failure is logged and the subordinate action is not attempted.
- If the primary action succeeds, the success is logged and the subordinate action is attempted.
  - If the subordinate action succeeds, the success is logged.
  - If the subordinate action fails, the failure is logged.

Sample Data File

The following sample data file consists of 1 header line and a 1 data line.

```
Action,LastName,FirstName,DefLogin,DefShell,TokSerial,ReplTokSerial,TokEnabled,
SetPin,GrpName,GrpDefLogin,GrpDefShell
AUT,Smith,John,Smithj,,,853618,,,1,1234,local,,,fred.securid.com
```

Input Parameter File

Input parameter files, *.ini, can be created and used instead of command-line arguments.

The .ini file references the data file, and can contain the Super Admin credentials, operational options, and output parameters required for software token assignment.

Note: Use of an .ini file prevents the capture of a command-line password in the Linux history file.

The file is a text file, consisting of command-line options.

- One or more spaces must separate the command-line option and any arguments. When statements with embedded spaces are included, the number of spaces between two items is not maintained. If this is an issue, include the statement on the command-line.
- Variables containing spaces must be enclosed in double quotes. For example, “E MMM dd HH:dd:ss z yyyy.”

Listing of a possible **example.ini**:

```
-i input.csv
--newlog
-m 2
-o goodOldLog
-p 30
-r /opt/rsa/am/amba/results/tokenListResults.txt
-a admin
-P changeIt123
--ctkip
--lic amba.lic
```

When an .ini file is first encountered, it is automatically encrypted. The encrypted version replaces the unencrypted version. If the unencrypted version cannot be encrypted and replaced, the process fails and **AMBulkAdmin** terminates with an error.

Important: There is no process to un-encrypt an encrypted .ini file. If an un-encrypted version is needed, make appropriate backup copies of the file before use.

Command Reject File

Each run of the AMBA utility produces a command reject file. Any previously existing reject file is overwritten unless the file is renamed, or a new file name/location is configured.

The default filename is **AMBulkAdminReject**. The extension is determined by the input file extension. The default file is created in the current directory. The file name and location are configurable through the --rej [command reject file] option.

The command reject file contains a copy of each input line that fails for any reason. A comment line, with the reason for failure and the line number of the associated input line, precedes each failed input line. If a primary action succeeds and the secondary action fails, the entire line is entered into this file.

To correct input errors, edit the file to correct any errors. In cases where a primary action succeeded and a secondary action failed, remove the primary action portion and correct the secondary portion of the action. Supply this edited file as input to the next update.

The following is an example listing from an **AMBulkAdminRejects** file:

```
action,DefLogin,LastName
// Line 2: Unknown Action field: asdfas
asdfas,xyz,xyz
// Line 3: Principal with userid already exists in the realm:admin
au,admin,admin
```

Results File

Each run of the AMBA utility produces a results file, used to store the results of the List and Multiple actions. This file is overwritten on each run.

The default filename is **AMBulkAdminResults.CSV**. The default file is created in the current directory. The file name and location are configurable through the **-r *results file*** option and the Change the Results File Name action. For more information, see -r [results file | stdout | stderr].

For a list of the valid results files input fields and their descriptions that are generated while running List and Multiple actions, see User Information Fields and Token Information Fields.

Log File

The AMBA utility does not produce a log file by default. Message information is printed to the standard output channel, and can be redirected using standard operating system conventions, **stderr** and **stdout**.

To specify the use of a log file, use the **-o *log file*** option. The AMBA utility creates and appends log information to the log file. For more information, see -o [log file | stderr | stdout].

The default filename is **AMBulkAdminLog.txt**. The default output file is created in the current directory. Log information for successive **AMBulkAdmin** sessions is appended to the file, unless --newlog has been specified.

Log message levels are set using the -m 0 | 1 | 2 | 3 option.

The **errMsg** type message is the result of application type errors, such as invalid file names and directories and command-line errors.

Success and **Failure** type messages are used to identify the final result of a action or a secondary action. These message types contain the line number of the associated input line.

If "API Return:" appears in a **Failure** type message, it indicates that the Authentication Manager API returned an error, and the error text follows this string. All other **Failure** type messages are returned by the AMBA utility.

An example of each of these message types is provided here:

The following is an example of a log with each message type:

```
BOJ    : 2018-10-18 17:35:13 - RSA AMBulkAdmin version 8.4.0.0.0; Input =
f:\input.csv
Info   :                               -Output Log File Opened
Info   : Line 1                        -Header Line
Info   :                               -Entering listUserInfoByField
Error  : Line 2 - listUserInfoByField  -CompareValue is required.
Failure: Line 2 - listUserInfoByField  -API return: CompareValue is required.
Info   :                               -Entering addUser
Success: Line 3 - addUser              -user1, user1LastName
Info   :                               -Leaving addUser
Info   :                               -Closing input file
Info   :                               -Closing rejected actions file
Info   :                               -Closing unsupported actions file
Info   :                               -Log File Closed
EOJ    : 2018-10-18 17:35:18 - Terminating
```

Add **--verbose** to the command-line to enable verbose logging, if required.

Software Token Database File

Software token database (SDTID) files are output database files for RSA SecurID (SecurID) software tokens assigned during **AMBulkAdmin** processing, using actions such as Add Token to User, Add Token to User Automatic, Add User and Token, Add User and Token Automatic, Single Softtoken Deployment, Replace Token, and Replace Token Automatic. By default, files are stored in the current directory.

The **-g** command-line option is required to instruct the AMBA utility to generate the software token database files. The **-gdir** command-line option can be used to place any generated files in the specified directory.

File names, in the format ***User ID_Token serial*.sdtid**, are based on the default login name and token serial number of the User, and have the extension .sdtid. For example, **juser_000027050105.stdid** where the default login name is *juser* and the assigned token serial number is *27050105*.

In Authentication Manager, each SDTID file can have a maximum of 100 tokens. For example, if the range provided is around 1000, then 10 software token database files are available in a zip file, 1 for each 100 tokens. The default zip file name is **softTokenDeployment.zip**.

Input File Templates

Input file templates can be used to create reusable input files. A template can be used as a base file for creating AMBA utility input files.

For more details see the --gtc templatename option.

The following is an example of a newly-created CSV template:

```
Action,IdentitySource,SecurityDomain,LastName,FirstName,DefLogin...
<data>,<data>,<data>,<data>,<data>,<data>...
// Replace <data> with actual data or delete it leaving "" then delete
// these comment lines
// The header labels or first line of this file can be deleted.
```

Email Template File

An XML template file can be used to construct email. The default template files included in the **/opt/rsa/am/utils/resources/amba_template_files** directory can be used as a starting point for building bespoke emails.

For more details, see Automatic Notification.

Action Codes

The following alphabetical reference table describes AMBA action codes:

Table 11: Action Codes

Action	Description
AAH	Add Agent Host
AG	Add Group
AGC	Add Group to Client
AP	Assign Profile to a User.
ARC	Add RADIUS Client
ARG	Add Remote Group
ARGC	Add Remote Group Client
ASD	Add Security Domain
ATU	Add Token to User
ATUA	Add Token to User Automatic
AU	Add User
AUG	Add User to Group
AUP	Add User and Password
AUR	Add User Remote
AURG	Add User to Remote Group
AUT	Add User and Token
AUTA	Add User and Token Automatic
CAU	Change or Add User
CAUP	Change or Add User and Password
CAUT	Change or Add User and Token
CIF	Change Input Format
CPAD	Change Principal Attribute Data
CPADC	Change Principal Attribute Data Clear

Table 11: Action Codes (Continued)

Action	Description
CPS	Change PIN Status
CRFN	Change the Results File Name
CTA	Change Token Attributes
CTD	Change Token on First Use of New Token
CTI	Change Token Immediately
CTS	Change Token Status
CTSD	Change Token Security Domain
CTSX	Change Token Status eXtended
CTU	Change Temporary User Mode status for a User.
CUR	Change User Remote
CUSD	Change User/Token Security Domain
DG	Delete Group
DGC	Delete Group from Client
DODA	Enable or Disable Risk-Based Authentication
DRG	Delete Remote Group
DRGC	Delete Remote Group Client
DSD	Delete Security Domain
DT	Delete Token from User
DU	Delete User
DUG	Delete User from Group
DURG	Delete User from Remote Group
EAFXD	Set Emergency Access Fixed
EAOFF	Set Emergency Access OFF
EAOTP	Set Emergency Access OTP
EODA	Enable OnDemand Authentication
ETL	Extend Software Token Lifetime

Table 11: Action Codes (Continued)

Action	Description
LSN	List Secondary Nodes for Agent Host
LTI	List Token Information
LTIF	List Token Information by Field
LTSR	List Token Summary Report
LUI	List User Information for User
LUIF	List User Information by Field
MSD	Multiple Softtoken Deployment
MTA	Multiple Token Assignment
MTD	Multiple Token Disable/Rescind
MTR	Multiple Token Replacement
QUIT	The quit action is used to terminate the standard input file, **STDIN**. It is ignored if present in a disk input file. There are no required or optional fields.
RBA	Enable or Disable Risk-Based Authentication
REPT	Replace Token
REPTA	Replace Token Automatic
RT	Rescind Token
SSD	Single Softtoken Deployment
SSTP	Set Software Token Profile
UAR	User Admin Role
UEU	Unregister External User
UODA	Update OnDemand Authentication
UP	Unassign Profile
UT	Unassign Token
UUD	Update User Data

Input Field Definitions

These definitions are general; however, specific actions can use some fields in a nonstandard manner in order to keep the number of fields to a reasonable level. The **SetPin** field is an example of such a definition. Although its intended use is to set PINs for related actions, it is also used to supply passwords for password-related actions.

Note: The special characters & % > < and ` are not allowed.

Table 12: Input Fields

Field	Description
AgentHostAddress	Used for the **AAH A**dd operation. If either **AgentHostName** or **AgentHostAddress** is used, then an address or name lookup is attempted for the other. If both fields are present, no lookups are performed and the values are forced.
AgentHostName	Used with the **AAH Add**/**Remove**/**Update** operations. Mandatory for the **Remove** and **Update** operations. For the **Add** operation, if either **AgentHostName** or **AgentHostAddress** is used, an address or name lookup is attempted for the other. If both fields are present, then no lookups are performed, and the values are forced.
AgentHostType	Used for the **AAH Add** and **List** operations, for adding a new **AgentHost** and filtering the results. Supported values are: 1 - Standard Agent 2 - Web Agent 3 - Radius Server Agent 4 - Radius Client Agent The default for the **Add** operation is1. If not specified for the **List** operation, all **AgentHostType** are retrieved.
AgentNewHostname	Used for the **AAH Update** operation. Updates the existing **AgentHostname** with a new value.
AgentRestriction	Used for the **AAH Add**/**List**/**Update** operation. User Group Access Restriction: 0 - Restriction is disabled 1 - Restriction is enabled For the **Add** operation, the default is 0.

Table 12: Input Fields (Continued)

Field	Description
AgentTrustRealm	Used with the **AAH Add** or **Update** operations to modify the Authentication Manager trusted realm setting for an Agent. Supported values are: 0 - Trusted realm authentication not enabled. 1 - Trusted realm authentication is enabled, and open to all trusted Users. 2 - Trusted realm authentication is enabled, but only trusted Users in trusted User Groups with access to the Agent can authenticate. The default is 0.
AnyRadiusClient	Used with the ARC command to add the ANY RADIUS client. The ANY client does not track which RADIUS client sends authentication requests. Authentication requests using the shared secret specified for the ANY client are processed regardless of the originating client's IP address.
AttributeName	Used to provide the name of a custom attribute, **AttributeValue**, defined under Token Attribute Definitions in Authentication Manager
AttributeValue	Used to provide a value for a custom attribute named in **AttributeName**. **AttributeName** and **AttributeValue** can be defined as numbered pairs, used to provide a value for a custom attribute named under the respective attribute name pair. The numbered pairs can be declared in any order, and the search for the attribute name is carried out in the applicable **IdentitySource**.
CertDN	User's certificate DN.

ChangePwdFlag	Used to force a password change when using the Add User actions. Supported values are **true** or **false**, the default. For example:
	`./rsautil AMBulkAdmin -a SuperAdminUserID -P SuperAdminPassword --verbose -m 0 -i au.csv`
	Where the **au.csv** file contains:
	`action,deflogin,lastname,UserPwd,ChangePwdFlag,EnableFlag`
	`au,User1,User1,Password$,true,true`
	`au,User2,User2,Password!,true,true`
	When the process is complete, each User is prompted to change their password during their first successful logon.

ClearNodeSecret	Used with the **AAH Update** operation. Allowed values are: **true** - clears the Node Secret associated with the **AgentHost.** **false** - does not clear the Node Secret (default).
ClntDefLogin	Client default login.
ClientIPAddress	Client IP address. Used by the ARC action to specify an IP address for a RADIUS client.
ClientIPAddressType	Used with the ARC command. Enter 0 or N for an IPv4 address. To add an IPv6 RADIUS client, you must use the Security Console, which allows you to add both IPv4 and IPv6 RADIUS clients.

Table 12: Input Fields (Continued)

Field	Description
ClntName	**AgentHostName** [Name of the Agent to register the Group on, maximum 256 characters]. Although both the fields represent the same entity, they are not interchangeable. This keeps AMBA aligned as closely as possible with ACEBulkAdmin. Used with the ARC command to add a RADIUS client.
CompareField	Used in the list actions to indicate which field to use as a filter and selector for report data selection. Consult the individual list action definitions for allowable entries for this field. The default is 0. If either **CompareType** or **CompareValue** is 0, **CompareField** is assumed to be 0.
CompareType	Used in the list actions to indicate what type of comparison to apply to the **CompareField**. Consult the individual list action definitions for allowable entries for this field. The default is 0. If either **CompareField** or **CompareValue** is 0, **CompareType** is assumed to be 0.
CompareValue	Used to supply values for list action filter/selector report data selection. Consult the individual list action definitions for allowable entries for this field. The default is 0. If either **CompareField** or **CompareType** is 0, **CompareValue** is assumed to be 0.
CopyProtect	Specifies whether the copy protection is enabled for the Token. • If 0, copy protection is disabled. • if any other value, copy protection is enabled. The Token record cannot be removed from the directory in which it is installed on a user's computer.C
CreateAssociatedAgent	Used with the ARC command to create the RADIUS client with an agent. You can configure RADIUS clients with or without an assigned authentication agent. The difference between the two methods is in the level of access control and logging you want to have. • RADIUS client with an agent. Adding an agent to a RADIUS client allows Authentication Manager to determine which RADIUS client is used for authentication and to save this information in log files. • RADIUS client without an agent. Without an assigned RADIUS client agent, Authentication Manager cannot track which RADIUS client sends authentication requests and you cannot assign a profile to the client.

Table 12: Input Fields (Continued)

Field	Description
CreatePin	Used to set the validity period of emergency access codes. The format is Dmm/dd/yyyyHnLn. Any Date, Hours, and Lifetime entry must appear in that order, DHL. For example, "D90H0" = 90 days, "D0H12" = 12 hours. The default is 14 days, from the current date and time. Days can range from 0 through 365 and hours can range from 0 through 23. The **L** component is optional and represents the number of hours until the emergency access mode expires. If the **L** component is greater than 0, the **D** and **H** components are ignored. Acceptable entries are: Dmm/dd/yyyy Dmm/dd/yyyyHn Dmm/dd/yyyyLn HnLn Ln An hour entry is not valid unless a date entry is present. HnLn is valid because when a Lifetime entry is present, the Date and Hour entries are ignored.
DefLogin	Specifies the User's default login or account name, when User is assigned to a Group where a Group-specific login name is not supplied. The maximum is 255 characters.
DefShell	Specifies the User's default shell, when a User is assigned to a Group where the Group-specific shell is not supplied. The maximum is 256 characters.
DeliveryMethod	Specifies the type of notification, SMTP or SMS. If only one delivery method is configured, the actions automatically select that method. If both delivery methods are configured, the delivery method to be used must be indicated.
DestinationAddress	Used for email addresses and SMS phone numbers, depending on delivery method. For email addresses, it overrides any existing principal email address.
DestinationAttributeName	Specifies a new Attribute Name.
DestinationSecurityDomain	Specifies the destination security domain.
DeviceserialIsCtkipCode	Specifies the **Deviceserial** field value used for the CTKIP Activation Code. This is applicable only when the **--ctkip** command-line option is declared.
DeviceSerialNumber	Provides a device serial number or User default login in a range, for software token activations.
Email	Specifies the User's email address
EnableFlag	**true** to enable a User or Agent account, **false** to disable the account.
EndRange	Declares the last Token serial number in the range of Tokens to be processed.

Table 12: Input Fields (Continued)

Field	Description
ExpiryDate	Date the OnDemand Authenticator automatically expires. Format is yyyy.mm.dd.HH.mm z
ExtnDataOption	Used to determine the inclusion of User extension data in the list. Can contain the following values: 0 - no additional data listed 1 - include Token extension data 2 - include User extension data 3 - option 1 and 2 4 - include Group membership Group names 5 - option 1 and 4 6 - option 2 and 4 7 - option 3 and 4 8 - include Group membership field values 9 - option 1 and 8 10 - option 2 and 8 11 - option 3 and 8 Consult the individual list action definitions for allowable entries for this field.
Filename	Used to specify or rename the Token output SDTID or zip file. See Software Token Database File.
FirstName	Specifies the User's real first name. The maximum is 255 characters.
ForceGroupSearch	A User can be added only to a Group in the same Identity Source or the Internal Database. If this variable is set to **true**, the search for the Group name is made in the Internal Database, otherwise the search s made in the User's IdentitySource.
GrpDefLogin	Specifies the User's login or account name for Agents that the specified Group is activated on. The maximum is 48 characters. If not supplied, the **DefLogin** value is used.
GrpDefShell	Specifies the User's shell for Agents that the specified Group is activated on. The maximum is 256 characters. If not supplied, the **DefShell** value is used.
GrpName	Specifies the name of a Group for a User or a RADIUS client. The maximum value is 255 characters.
IdentitySource	Specifies the Identity Source where a search is made. The default is the identity source mapped to the default security domain.
InstanceName	Specifies the name of the target instance when multiple instances have been configured.
Key	Used for search and lookup arguments.
KeyType	Specifies the type of search or lookup argument in **Key**.
LastName	Specifies the User's real Surname or Family name. The maximum is 255 characters.

Table 12: Input Fields (Continued)

Field	Description
Limit	Used to limit the number of entities, such as Tokens or Users to be actioned. This allows some control over the size of the task. A Limit of 0 or empty implies no limit or "action all entities." The value, when present, must be a positive integer.
MinTokenLife	Used to guarantee that a Token has a minimum number of days before it expires. The search for a Token looks for Tokens with an expiry date greater than the current date. Adding a number of days with this field adjusts the search to Tokens with an expiry date greater than **today** + **MinTokenLife**. This field is ignored if Token Type is set to 'Next available Token'.
MiscVariable	Used to supply miscellaneous information to various actions. The definition of the contents of this field can be found in the specific action descriptions where it is declared.
Nickname	Used by Token assignment actions to supply a value to the Token nickname when applicable.
NicknameIsCtkipCode	Used to indicate the **Nickname** field value is used for the CTKIP activation code. This is applicable only when the **--ctkip** command-line option is declared.
OutputOption	Specifies the formatting and other options to the list actions. It determines whether a header line and extended User fields should be output to the list. Consult the individual list action definitions for allowable entries for this field. For a full list of values, refer to OutputOption Values.
OverOption	Specifies the SDTID file overwrite option. If any value other than 0, the file is overwritten; If 0 is used and if the file already exists, the content will be appended to the existing zip file, otherwise a new zip file will be created with the name provided.
ParentDomainName	Specifies the name of the parent security domain for the security domain to be added or deleted.
Password	Specifies that a password be provided by an administrator in order to access a software token SDTID file. If there is no password associated with the file, an empty string may be passed.
PINIndicator	Specifies the type of PIN and PIN mode assignment. Acceptable values or this action are: **SET_TEMP_PIN** - the ODA is initialized with the provided PIN. Authenticator is in New PIN Mode. **SET_PERM_PIN** - the ODA is initialized with the provided PIN. Authenticator is not put into New PIN Mode. **GENERATE_PIN** - ODA is initialized with a system-generated PIN. Authenticator is not put into New PIN Mode. PIN format is determined by the User's token policy. **NO_PIN_UPDATE** - the User's PIN is not modified.

Table 12: Input Fields (Continued)

Field	Description
PinMode	Used to determine if a New PIN Mode should be set. 0 - Maintain the existing pin mode has an effect only if the Token already has a PIN (default). 1 - Set the newly assigned Token to New PIN Mode. For more information on the **PinMode** field, see Change PIN Status.
PinType	Used to indicate a passcode (PIN + tokencode) or tokencode Token.
PolicyName1 thru 5	Specifies the policy name to be selected for the matching **PolicyType** in the **ASD** action.
PolicyType1 thru 5	Specifies the policy type for policy selection in the **ASD** action. Entries are case-sensitive. **Note:** Each **PolicyType** must have a matching **PolicyName** entry. Valid values are: • PasswordPolicy • LockoutPolicy • SelfServicePolicyAM_Token_Policy • AM_OFFLINE_AUTHN_POLICY
ProfileName	Specifies the profile name during User profile maintenance.
RadiusClientModel	Specifies the type or manufacturer of a RADIUS client for the ARC action, for example, “Cisco ASA Firewall” or “IP3 Networks.”
RangeMode	Specifies criteria used to deploy assigned software tokens either by serial number or user default login. Supported values are 0, 1, 2, 3, and 4.
RBAOperation	Specifies one of the following operations for each User: • Enable - Enables the User for RBA. • Disable - Disables the User for RBA. • ClearSQ - Clears the User’s set of security questions. • ClearDevice - Clears the device mapping for the User.
RealmName	Specifies the realm name for remote User maintenance.
RegenerateSeed	Forces the generation of new tokencodes when set to **true**, or maintains the current tokencodes, when set to **false**. The default value is **true**.
RemoteAlias	Specifies a remote login User name.
RemoveGrpDefLogin	Removes a logon alias from a group. Used with the **CAU** command.
ReplTokSerial	Specifies a Token serial number, up to 12 numeric characters (0-9). Leading zeros are optional.
RoleName	Specifies the Administrative Role.
RoleSecurityDomain	Specifies the Security Domain for the Role. The default is the System Domain.

Table 12: Input Fields (Continued)

Field	Description
SecurityDomain	Specifies the Security Domain in which the search for the relevant entity is made. The default is the System Domain.
SecurityDomainCreatedBy	Specifies the **CreatedBy** entry for added security domain
SecurityDomainDescription	Specifies the description of Security Domain.
SecurityDomainName	Specifies the name of the security domain to add or delete.
SetPin	Sets or clears the PIN status. The default is 0. • C, c or 0 = New PIN Mode, PIN Cleared. • N = New PIN Mode, Old PIN Required, not valid in an Add action. Any other string value attempts to set the PIN to that value. System PIN rules determine the success of setting the PIN. Also used to supply a password for the **AUP** and **CAUP** actions. For the emergency access actions, this field is used to supply a password, and it can be used with the following format to set the number of passwords to generate. • Nn where n is the number of OTP to issue. In Authentication Manager, the Length and Format of OTPs are set by token policies and cannot be set manually through a command. For more information on the **SetPin** field, see Change PIN Status. Check the token policies in Authentication Manager before using this field.
SharedSecret	Specifies the authentication shared secret (case-sensitive password) used by a RADIUS client.
SiteFile	Specifies the URL for the software token site URL LIST during token activation
SiteURL1	Specifies the URL for the software token site 1 during token activation.
SiteURL2	Specifies the URL for the software token site 2 during token activation.
SiteURL3	Specifies the URL for the software token site 3 during token activation.

Table 12: Input Fields (Continued)

Field	Description
SoftIDParams	Used by the add and delete Token actions when distributing tokens using SDTID files. This field must contain three decimal digits that control the following seed file generation characteristics: First digit: 0 - required but ignored. Second digit - Copy Protection Flag: 0 - Copy protection off. 1 - Copy protection on. Third digit - Password usage and Interpretation method: 0 - No password. 1 - Static password. See **SoftIDPW** below. 2 - Default login. 3 - Default login appended to static password. Not required when using CT-KIP.
SoftIDPW	Used by the add/replace Token actions. Specifies a password to be used for the seed file encryption when **SoftIDParams** specifies a static password.
SoftTokenProfile	Specifies a software token profile name. This name must be identical to the software token profile name that is defined in the Security Console. For examples of the use of this field, see Software Token Distribution.
StartRange	Specifies the beginning software token serial number or user default login in a range. If **RangeMode** is set to 1, this argument is ignored.
SubCommand	Used to identify the specific operation to be performed on an attribute.
SubDomain	Used to indicate whether or not a security domain search should include sub-domains. The supported values are: 0 - Do not include sub domains 1 - Include sub domains. The default for Change actions is 0. The default for Delete and RBA actions is 1.
TemplateFile	Specifies the XML template file to use to construct email notifications. If empty, a built-in template is used. The default template files included in the **/opt/rsa/am/utils/resources/amba_template_files** directory can be used as a base for building your own.
TokEnabled	Specifies the post-assignment Token status. 0 = disabled and 1 = enabled. All unassigned tokens are disabled by default. when a token is assigned, it is automatically enabled unless TokEnabled is set to 0. This action affects only the specified Token.
TokenAssigned	Used to indicate whether the list of tokens to be moved are Assigned or Unassigned. By default all the unassigned tokens are searched. The supported values are: 0 - Unassigned (default) 1 - Assigned Tokens

Table 12: Input Fields (Continued)

Field	Description
TokenSerial	TokenSerial determines the replacement action: 0 - unconditional replacement (default) 1 - replace if expired Token is the only assigned Token.
TokenType	The variable is used to limit the token lookup. If not specified all types of token is searched. The supported values are: 0 SecurID Standard Card 1 SecurID PINPAD Card 2 SecurID Key Fob 3 SecurID Watch 4 SecurID software token 5 SecurID Smartcard 6 SecurID Modem 7 SecurID Crypto 8 SecurID Proteus 9 SecurID USBCOSMO (SID800) 10 SecurID Flextoken
TokSerial	Specifies the Token serial number, up to 12 numeric characters (0-9). Leading zeros are optional.
UserPwd	IMS Password for each individual User. This field takes precedence over the **--userPwd** command-line option which provides a single password to all applicable Users.
UseSameSecurityDomain	Used with the **ETL** command. Requires AMBA to only select unassigned software tokens that share the same security domain as the tokens that are being extended.

The following input fields and their descriptions are valid in results files that are generated while running **List** and **Multiple** actions.

Table 13: User Information Fields

Field	Description
chLastName	User's last name
chFirstName	User's first name
chDefaultLogin	User's default login
chSecurityDomain	User's security domain
chDefaultShell	User's default shell
bTempUser	Indicates whether User is a temporary User (TRUE/FALSE)
dateStart	Start date for temporary User. Stored as Coordinated Universal Time (UTC)
dateEnd	End date for temporary User. Stored as UTC.

Table 14: Token Information Fields

Field	Description
chSerialNum	Token serial number
chSecurityDomain	User's security domain
iInterval	Number of seconds between display changes
dateBirth	Date the Token was activated
dateDeath	Date the Token shuts down
dateLastLogin	Date of the last login with this Token
iType	Token type. See TokenType.
bHex	Indicates whether the display is hexadecimal (TRUE/FALSE)
bEnabled	Indicates whether the Token is enabled (TRUE/FALSE)
bNewPINMode	Indicates whether the Token is in New PIN Mode (TRUE/FALSE)
bMustCreatePIN	Indicates whether User must create PIN (TRUE/FALSE)
iNextCodeStatus	Next tokencode status: 0 Not in the next tokencode mode 1 Token is in the next tokencode mode
iBadTokenCodes	Number of bad tokencodes entered
datePIN	Date PIN was last changed
dateEnabled	Date Token was last enabled or disabled
dateCountsLastModified	Date Token counts were last modified
Note: The following fields apply to SecurID software tokens only. For all other tokens, the fields are filled with zeros	
bProtected	Whether software token was copy-protected on last deployment: 0 No 1 Yes
bDeployed	Indicates whether a software token is currently deployed: 0 No 1 Yes
iCount	Number of times the Token has been deployed
ExtnKey	Attribute Name of the User/Token being requested.
ExtnData	Attribute Value of the User/Token being requested.

Table 15: Software Token Device Type Fields

Field	Description
chSTDTDescription	Description field.
chSTDTFamilyKey	The family key can be a resource bundle key or the full product family description of the device type. The family key plus the version is considered a unique key for identifying this software token device type.
chSTDTLabelKey	Label Key
chSTDTPluginModuleName	The plug-in module name is used for Token exporting. This can be customized plug-ins, such as the TSF plug-in, or one of two system-embedded plug-ins: • PC Software Token V2.4 or earlier for 64-bit Tokens • PC Software Token V3.0 or later for 128-bit Tokens.
chSTDTVersion	Version.
bSTDTIsPinpad	Pin type is pinpad, true or false.
chSTDTTokenCodeLength	Tokencode length, 6, 8, 6|8, 8|6
chSTDTTokenCodeInterval	Tokencode interval 30, 60, 30|60, 60|30
chSTDTTokenCodeType	Tokencode type, time or event

Table 16: OutputOption Values

Value	Description
-31	List Token serial numbers and CT-KIP activation codes, download URLs and additional CT-KIP related information. Only lists data for CT-KIP capable Tokens.
-21	List Token serial numbers and CT-KIP download URLs. Only lists data for CT-KIP capable Tokens.
-11	List Token serial numbers and CT-KIP activation codes. Only lists data for CT-KIP capable Tokens.
-2	List default logins only and any assigned Token serial numbers (no data).
-1	List Token serial numbers/default logins only (no data).
0	List Token information/User information.
1	Append LDAP source to option 0.
2	Append remote alias to option 0.
3	Append profile name to option 0.

Table 16: OutputOption Values (Continued)

Value	Description
4	Combine options 0, 1, 2, and 3.
5	List LDAP source only.
6	List remote alias only.
7	List Profile name only.
10	List Base Information + *UserInfo*.
11	Append LDAP source to option 10.
12	Append remote alias to option 10.
13	Append profile name to option 10.
14	Combine options 10, 11, 12, and 13.
15	List LDAP source only.
16	List remote alias only.
17	List profile name only.
20	List both Token and User information.
21	Append LDAP source to option 20.
22	Append remote alias to option 20.
23	Append profile name to option 20.
24	Combine options 20, 21, 22, and 23.
25	List Token information and LDAP source.
26	List Token information and remote alias.
27	List Token information and profile name.
30	Append option 0 to Token info for assigned Tokens.
31	Append option 1 to Token info for assigned Tokens.
32	Append option 2 to Token info for assigned Tokens.
33	Append option 3 to Token info for assigned Tokens.
34	Append option 4 to Token info for assigned Tokens.
60	Append option 30 to replacement information for the assigned Tokens.

Table 16: OutputOption Values (Continued)

Value	Description
A | a	Append output to an existing results file.
E | e	Extract files.
H | h	Write a header record to the results file.
N | n	Notify. Send an SMTP message with a generated PIN to principal. Message type depends on DeliveryMethod.
O | o	Omit the PIN from the results file. PIN is replaced by 'omitted'.
R | r	Report only.
T | t	Change the file name to ***Token serial*.sdtid.**
U | u	Change the file name to ***User ID*.sdtid**.

3 Software Token Notifications and Attributes

The topics in this chapter include:

- Automatic Notification
- Software Token Device Type Attributes

Automatic Notification

When provisioning software tokens, the AMBA utility can automatically send email notifications to Users. The following AMBA utility actions include fields that support this feature:

- Add User and Token
- Add User and Token Automatic
- Add Token to User
- Add Token to User Automatic
- Single Softtoken Deployment
- Change or Add User and Token
- Replace Token.

Automatic notifications can deliver the following information:

- For CT-KIP provisioning, the activation code and delivery URL.
- For Software Token Database File provisioning, the .sdtid file (SDTID).

  There are two delivery methods for SDTID files, SMTP and SMTP2. Both methods generate an email with the Token SDTID file as an attachment, however SMTP2 will generate a follow-up email with the SDTID file password. If the delivery method is SMTP2 and the SDTID file is not password protected, an error is thrown and the SDTID file is not sent.

The **CAUT** action uses automatic email notifications solely to deliver CT-KIP credentials. The other actions that support automatic notifications can deliver CT-KIP credentials, SDTID files, and SDTID file passwords.

For automatic notification, the RSA Authentication Manager instance must be configured with an SMTP mail server. This can be done through the Security Console **Setup > Instances** tab. Select the drop-down menu for the Instance and select **Mail Server (SMTP)**.

System Emails

The Automatic Notification System is triggered when the **DeliveryMethod** field of a Token-related action is set to either *SMTP* or *SMTP2* and the AMBA utility has been initiated with either the **-g** or **--ctkip** command-line option. Leave the **DeliveryMethod** field empty to disable notification.

For email delivery set this field to one of the following values: .

Provisioning Type	DeliveryMethod	Description
CT-KIP	SMTP	Email activation code and URL
SDTID file	SMTP	Email SDTID file
SDTID password protected file	SMTP2	Email SDTID file followed by separate email of password
any	*empty*	Disables automatic notification

If the Token action is successful, then the appropriate email is sent.

InstanceName provides the name of the current instance. This is required only if **DeliveryMethod** is set to SMTP and multiple instances have been declared for this server. The **CAUT** action does not use this field.

By default, the User's account email address is used, however this can be overridden by including **DestinationAddress** with the Token action and setting it to the desired email address.

TemplateFile is used to provide the path and file name of an email template file. Providing a customized template allows additional text to be included in the email along with changes in format. The default template files included in the **/opt/rsa/am/utils/resources/amba_template_files** directory can be used as a starting point for building your own.

The AMBA utility has a number of built-in email templates for the various possible notifications. The built-in types are for the most part copies of the provided template files. If the **TemplateFile** field is empty, the AMBA utility uses a built-in template.

If a template name contains multiple dot (.) separators, the last occurrence defines the suffix.

The Javamail system automatically converts separators (\ and /) to double underscores (__) in attachment file names. Therefore an SDTID file named **c:\temp*Userid_Tokenserial*.sdtid** is named **c__temp__*Userid_Tokenserial*.sdtid** as the email attachment name.

To Customize a Template:

There are a few restrictions that should be understood before customizing a template.

The "DO NOT MAKE ANY CHANGES TO DOCTYPE" in each template file is true with one exception. You can add or delete "ENTITY" statements from the "DOCTYPE" section. Additions must be formatted exactly like the existing statements. Order is not important in this section.

You can add new ENTITY names, but they must be from the following list:

<!ENTITY uid ""> (User's default logon)

<!ENTITY fnm ""> (User's first name)

<!ENTITY lnm ""> (User's last name)

<!ENTITY tok ""> (Token serial)

<!ENTITY pwd ""> (SDTID file password)

<!ENTITY fil ""> (SDTID file name)

<!ENTITY PIN ""> (oda pin)

<!ENTITY acd ""> (CT-KIP activation code)

<!ENTITY url ""> (CT-KIP activation url)

After an ENTITY is declared in DOCTYPE, its "place holder" can be entered in any *text* statement and any number of times. Place holders are formed by wrapping an ENTITY name with "&;". For example, the place holder for the ENTITY name **uid** is **&uid;.** Any place &uid; appears in text it is replaced by the User's default logon when the email is generated.

Not all ENTITY names can be used in all templates. For example, PIN is applicable to On-Demand notices only. Although PIN does not produce an error on a Token provisioning notice, it is empty.

If a custom template is provided for SMTP2, then two files must be provided. The first file can be named with any legal name the operating system will accept and is used for the Token file attachment. A second file must be provided with the exact name as the first template except it will have "PW" appended to the name, not the suffix. This template is used for the password notification. For example:

Token attachment template
custom_template.xml
custom_template

Password notification template
custom_templatepw.xm
custom_templatepw

Software Token Device Type Attributes

Software token device types contain various sets of attributes which are basically name and value pairs. A specific software token device type can be linked to a software token through the Security Console, the API, or the Set Software Token Profile action. Default values can be assigned to these attributes.

Use the following actions to define values for software token device type attributes:

- Add User and Token
- Add User and Token Automatic
- Add Token to User
- Add Token to User Automatic
- Single Softtoken Deployment
- Change or Add User and Token
- Replace Token
- Multiple Softtoken Deployment
- Multiple Token Assignment
- Multiple Token Replacement.

Use the variables declared in the following table to assign values to the attributes for specific tokens.

Software Token Device Type Attributes	Field Name	Permitted Values
Nickname	Nickname	-2, -1, 0, value, empty
DeviceSerialNumber	DeviceSerialNumber	-2, -1, 0, value, empty
TOOLBAR_SITEFILE_URL	SiteFile	0, value, empty
TOOLBAR_SITEURL1	SiteURL1	0, value, empty
TOOLBAR_SITEURL2	SiteURL2	0, value, empty
TOOLBAR_SITEURL3	SiteURL3	0, value, empty

Where the permitted attribute values have the following meaning:

-2	copy the **TokSerial** to the attribute value field
-1	copy the **DefLogin** to the attribute value field
0	force the attribute value field to empty (overrides any default)
value	copy value to the attribute value field (overrides any default)
empty	use software token device type value if one is declared.

4 Add Actions

This chapter provides a description of the Add actions, and details the required and optional fields.

If software tokens are assigned, the **-g** and **-gdir** Command-line Options can be used to instruct the AMBA utility to generate Software Token Database Files and place them in the specified directory. If the **-g** or **--ctkip** options are not used, these actions will assign Tokens but will not build any output files or generate any CT-KIP credentials. The Set Software Token Profile action can be used to force a specific device type for credential generation.

Where fields are not described, see the description provided in the Input Field Definitions.

The topics in this chapter include:

Add User

Action	**AU**
Required Fields	LastName, DefLogin
Optional Fields	FirstName, Email, CertDN, DefShell, GrpName, GrpDefLogin, GrpDefShell, UserPwd, ChangePwdFlag, IdentitySource, SecurityDomain, EnableFlag, ForceGroupSearch, AttributeName, AttributeValue, AttributeName1, AttributeValue1, AttributeName2, AttributeValue2, AttributeName3, AttributeValue3, AttributeName4, AttributeValue4

Adds a new User and optionally adds the User to an existing Group. A User can be added only to a Group in the same identity source. The Group is not required to be present in the same domain as the User, but must be under the same realm.

UserPwd specifies the User password. If no password is provided and **IdentitySource** is Internal Database, Users are added with the password set as null. If the **IdentitySource** is an external directory, such as an LDAP server, the action throws an exception.

ChangePwdFlag forces a User password change. Supported values are **true** or **false**, the default.

For example:

```
./rsautil AMBulkAdmin -a SuperAdminUserID -P SuperAdminPassword
--verbose -m 0 -i au.csv
```

Where the **au.csv** file contains:

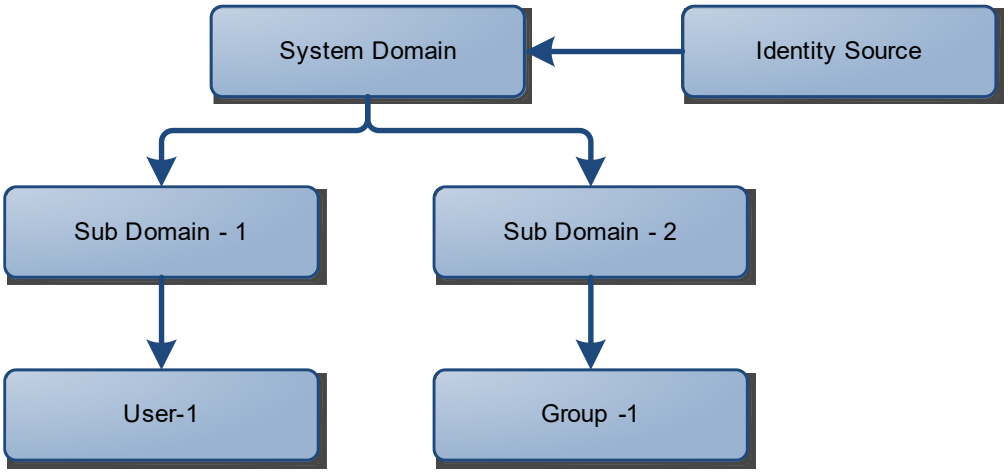
```
action,deflogin,lastname,UserPwd,ChangePwdFlag,EnableFlag
au,User1,User1,Password$,true,true
au,User2,User2,Password!,true,true
```

When the process is complete, each User is prompted, during their first successful logon, to change their password.

If **IdentitySource** and **SecurityDomain** are not provided, then the User is added in the default identity source, Internal Database, and security domain, System Domain created during RSA Authentication Manager (Authentication Manager) installation.

If **ForceGroupSearch** is set to **true**, the search for the Group name is made in the Internal Database, otherwise the search is made in the User's **IdentitySource**.

The following illustration is an example of a security domain.




```
Action,LastName,DefLogin,FirstName,DefShell,GrpName,GrpDefLogin,
GrpDefShell,UserPwd,IdentitySource,SecurityDomain
au,Scott,User-1,Tiger,,Group-1,,,password$,,Sub Domain-1
```

In the above example, User-1 is under Sub Domain-1, while Group-1 is under Sub Domain-2. In the input, the domain to which the User should be assigned is mentioned: Sub Domain-1. This action finds Group-1, by identifying the parent domain of User-1, which in this case is System Domain, and then uses this as the base domain to search for Group-1.

Add User and Token

Action	**AUT**
Required Fields	LastName, DefLogin, TokSerial, TokEnabled
Optional Fields	FirstName, Email, CertDN, DefShell, GrpName, GrpDefLogin, SetPin, PinMode, PinType, GrpName, GrpDefLogin, GrpDefShell, FileName, SoftIDParams, SoftIDPW, UserPwd, IdentitySource, SecurityDomain, SoftTokenProfile, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, RegenerateSeed, NicknameIsCtkipCode, DeviceserialIsCtkipCode, EnableFlag, ForceGroupSearch, DeliveryMethod, DestinationAddress, TemplateFile, InstanceName, AttributeName, AttributeValue, AttributeName1, AttributeValue1, AttributeName2, AttributeValue2, AttributeName3, AttributeValue3, AttributeName4, AttributeValue4

Adds a User and assigns the Token specified by **TokSerial**. The Token is enabled, unless **TokEnabled** is set to 0, the PIN is cleared, and both **BadTokenCodes** and **BadPINs** are set to zero. The specified Token and Group are not required to be present under the same domain as the User, but must be under the same realm. The User can be assigned to an existing Group. If the User has previously been added, a **Failure** message is generated.

If the **-g** or **--ctkip** options are not used, this action assigns Tokens, but does not build any output files or generate any CT-KIP credentials. For more information, see Software Token Database File and Command-line Options.

SoftIDParams is **required** if SDTID files are used for token seed delivery, and n**ot required** when using CT-KIP. It must contain three decimal digits to control seed file generation characteristics:

First digit:

0 - Required but ignored.

Second digit - Copy Protection Flag:

0 - Copy protection off.

1 - Copy protection on.

Third digit - Password usage and Interpretation method:

0 - No password.

1 - Static password. See **SoftIDPW** below.

2 - Default login.

3 - Default login appended to static password.

SoftIDPW specifies a password. To be used **only** when **SoftIDParams** specifies a static password.

UserPwd specifies the User password. If no password is provided and **IdentitySource** is Internal Database, Users are added with the password set as null. If the **IdentitySource** is an external directory, such as an LDAP server, the action throws an exception.

If **IdentitySource** and **SecurityDomain** are not provided, the User is added in the default identity source, Internal DataBase, and security domain, System Domain.

SoftTokenProfile specifies the software token profile name. If not provided, the User is added and associated with a Token, but the Token is not deployed.

Nickname, **DeviceSerialNumber**, **SiteFile**, **SiteURL1**, **SiteURL2**, and **SiteURL3** assign values to Software Token Device Type Attributes. A specific software token device type can be linked to a software Token through the Security Console, the API, or the Set Software Token Profile action.

RegenerateSeed forces the generation of new tokencodes when set to **true,** or maintains the current tokencodes, when set to **false**. The default value is **true**.

If **ForceGroupSearch** is set to **true**, the search for the Group name is made in the Internal Database, otherwise the search is made in the User's **IdentitySource**.

DeliveryMethod, **DestinationAddress**, **TemplateFile**, and **InstanceName** are used for automatic email notification during CT-KIP and software Token database file provisioning. For more information, see Automatic Notification.

Add User and Token Automatic

Action	**AUTA**
Required Fields	LastName, DefLogin, TokEnabled, MiscVariable
Optional Fields	FirstName, Email, CertDN, MinTokenLife, FileName, DefShell, SetPin, PinMode, PinType, GrpName, GrpDefLogin, GrpDefShell, SoftIDParams, SoftIDPW, UserPwd, IdentitySource, SecurityDomain, SoftTokenProfile, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, RegenerateSeed, NicknameIsCtkipCode, DeviceserialIsCtkipCode, EnableFlag, ForceGroupSearch, DeliveryMethod, DestinationAddress, TemplateFile, InstanceName, AttributeName, AttributeValue, AttributeName1, AttributeValue1, AttributeName2, AttributeValue2, AttributeName3, AttributeValue3, AttributeName4, AttributeValue4

This action automates the Add User and Token action by obtaining an unassigned Token of a specified type from the system and calling the **AUT** action using the newly acquired Token serial number.

This action is identical to the **AUT** action except that you supply a token type is specified, instead of the token serial number and the AMBA utility attempts to find a non-expired, unassigned RSA SecurID (SecurID) Token of the requested type.

MiscVariable is used to supply the desired token type. Acceptable values are:

-1	First available unassigned Token	5	SecurID Smartcard
0	SecurID Standard Card	6	SecurID Modem
1	SecurID PINPAD Card	7	SecurID Crypto
2	SecurID Key Fob	8	SecurID Proteus
3	SecurID Watch	9	SecurID USBCOSMO (SID800)
4	SecurID software token	10	SecurID Flextoken

MinTokenLife is used to guarantee that a token has a minimum number of days before it expires. It is ignored if **MiscVariable** is set to -1.

If this action is successful, the newly acquired token serial number is inserted in the **TokSerial** field of the **AUT** action, which then is called. If that action is successful, the newly assigned Token serial number is reported in the AMBA utility transaction log.

Add User and Password

Action	**AUP**
Required Fields	LastName, DefLogin, SetPin
Optional Fields	FirstName, Email, CertDN, DefShell, GrpName, GrpDefLogin, GrpDefShell, UserPwd, IdentitySource, SecurityDomain, EnableFlag, ForceGroupSearch, AttributeName, AttributeValue, AttributeName1, AttributeValue1, AttributeName2, AttributeValue2, AttributeName3, AttributeValue3, AttributeName4, AttributeValue4

Adds the User and assigns a static password tokencode to the User. If the User already exists, a FAILURE message is generated.

Each User can be provided with an individual IMS password using the **UserPwd** field or the command-line option **--userpwd** can be used to provide a single password for all the Users. If **IdentitySource** is Internal Database and no password is provided, Users are added with an IMS password set as null.

The User can be assigned to an existing Group. If specified, the Group is not required to be present under the same domain as the User, but must be under the same realm.

SetPin specifies the User's initial password. The value becomes the User's static password tokencode. It should not be confused with the IMS[1] password defined in the **UserPwd** note.

UserPwd is used to add a User to the IMS layer.

If **IdentitySource** and **SecurityDomain** are not provided, the User is added in the default realm.

If the **IdentitySource** is an External Directory, like an LDAP server, this action throws an exception.

If **ForceGroupSearch** is set **true**, the search for the Group name is carried out in the Internal Database, otherwise the search s carried out in the User's **IdentitySource**.

[1]The IMS layer is RSA low level software that provides a database layer for multiple products. Once an AMBA account is created, the IMS password is very rarely used for Authentication Manager functions, and it is not related to the Authentication Manager static passcode, if one is declared.

Add User Remote

Action	**AUR**
Required Fields	DefLogin, RemoteAlias, RealmName
Optional Fields	DefShell, GrpName, IdentitySource, SecurityDomain, ForceGroupSearch

Adds a remote User to the database. **RealmName** specifies the realm name for remote User maintenance.

If a Group is specified, it not required to be present under the same domain as the User, but must be under the same realm.

If **IdentitySource** and **SecurityDomain** are not provided, the User is added in the default realm.

IdentitySource must be provided if the **SecurityDomain** is specified and is mapped to a different realm other than the default.

If **ForceGroupSearch** is set **true**, the search for the Group name is carried out in the Internal Database, otherwise the search is carried out in the User's **IdentitySource**.

Add Remote Group

Action	**ARG**
Required Fields	GrpName
Optional Fields	SecurityDomain

The **ARG** (Add Remote Group) command adds user groups from a trusted realm into the RSA Authentication Manager internal database.

If Security Domain is not provided, then the remote group is added in the default security domain (System Domain) created during RSA Authentication Manager installation.

A trusted user group restricts access to an agent that is enabled for trusted realm authentication. When you create a trusted user group and enable associated agents, only members of the trusted user group can access that authentication agent.

By adding a trusted user group, only users who have a business need to access the resources protected by the agent can authenticate. For example, by creating a trusted user group for human resource workers, you can limit access to personnel records to those in the group.

For example, a sample CSV file, called **arg.csv**, can contain the following data:

```
Action,GrpName
arg, tgroup1
arg, tgroup2
arg, tgroup3
```

After you run the following AMBA command, the three user groups listed above are added to the RSA Authentication Manager internal database:

```
./rsautil AMBulkAdmin -a admin -P Password$ --verbose -m 0 -i
arg.csv
```

Add User to Remote Group

Action	**AURG**
Required Fields	GrpName, DefLogin
Optional Fields	SecurityDomain

The **AURG** (Add User to Remote Group) command links a trusted user to a trusted user group in the RSA Authentication Manager internal database.

If Security Domain is not provided, then the remote group is added in the default security domain (System Domain) created during RSA Authentication Manager installation.

A trusted user group restricts access to an agent that is enabled for trusted realm authentication. When you create a trusted user group, only members of the trusted user group can access the agent that is enabled for trusted realm authentication.

You can add new trusted users to an existing trusted user group.

For example, a sample CSV file, called **aurg.csv**, can contain the following data:

```
Action, GrpName, DefLogin, SecurityDomain
aurg,tgroup1,tuser1
aurg,tgroup1,tuser2
aurg,tgroup2,tuser3
```

After you run the following AMBA command, the three trusted users listed above are added to trusted user groups in the RSA Authentication Manager internal database:

```
./rsautil AMBulkAdmin -a admin -P Password$ --verbose -m 0 -i
aurg.csv
```

Add Token to User

Action	**ATU**
Required Fields	DefLogin or TokSerial, ReplTokSerial, TokEnabled
Optional Fields	SetPin, PinMode, PinType, SoftIDParams, SoftIDPW, IdentitySource, SecurityDomain, SoftTokenProfile, FileName, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, RegenerateSeed, NicknameIsCtkipCode, DeviceserialIsCtkipCode, DeliveryMethod, DestinationAddress, TemplateFile, InstanceName

Assigns an unassigned Token, **ReplTokSerial**, to a User associated with the specified assigned Token, **TokSerial**, or **DefLogin**.

If both the **DefLogin** or **TokSerial** fields are provided, then **DefLogin** is given precedence over **TokSerial**.

- If **DefLogin** is provided, and if **IdentitySource** and **SecurityDomain** are not provided, then the search for the User is carried out in the default realm.
- If **TokSerial** is specified, **IdentitySource** and **SecurityDomain** values are not required and are ignored, if present.

The token identified by **ReplTokSerial** is not required to be present under the same domain as the User, but must be present in same realm.

If the User already has three Tokens assigned, then a **Failure** message is generated.

SoftIDParams is **required** if SDTID files are used for token seed delivery, and n**ot required** when using CT-KIP. It must contain three decimal digits to control seed file generation characteristics:

First digit:

0 - Required but ignored.

Second digit - Copy Protection Flag:

0 - Copy protection off.

1 - Copy protection on.

Third digit - Password usage and Interpretation method:

0 - No password.

1 - Static password. See **SoftIDPW** below.

2 - Default login.

3 - Default login appended to static password.

SoftIDPW specifies a password. To be used **only** when **SoftIDParams** specifies a static password.

Nickname, **DeviceSerialNumber**, **SiteFile**, **SiteURL1**, **SiteURL2**, and **SiteURL3** can be used to assign values to software token device type attributes. For more information, see Software Token Device Type Attributes.

DeliveryMethod, **DestinationAddress**, **TemplateFile**, and **InstanceName** are used for automatic email notification during CT-KIP and SDTID file provisioning. For more information, see Automatic Notification.

Add Token to User Automatic

Action	**ATUA**
Required Fields	DefLogin or TokSerial, TokEnabled, MiscVariable
Optional Fields	SetPin, PinMode, PinType, MinTokenLife, SoftIDParams, SoftIDPW, IdentitySource, SecurityDomain, SoftTokenProfile, FileName, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, NicknameIsCtkipCode, DeviceserialIsCtkipCode, DeliveryMethod, DestinationAddress, TemplateFile, InstanceName

Automates the Add Token to User action by obtaining an unassigned Token of a specified type from the system and calling the **ATU** action using the newly acquired Token serial number.

This action is identical to the **ATU** action except that you supply a token type instead of the replacement token serial number, and the AMBA utility finds an unassigned Token of the requested type. If successful, the newly acquired Token serial number is inserted in the **ReplTokSerial** field of the **ATU** action which is subsequently called. If that action is successful, the newly assigned Token serial number is reported in the AMBA transaction log.

The search for Tokens to be assigned is carried out in the same realm as that of the User. If the first available unassigned Token is requested, **MiscVariable = -1**, then the Authentication Manager API gets the Token that expires first, regardless of Security Domain.

MiscVariable is used to supply the desired token type. Acceptable values are:

-1	First available unassigned Token	5	SecurID Smartcard
0	SecurID Standard Card	6	SecurID Modem
1	SecurID PINPAD Card	7	SecurID Crypto
2	SecurID Key Fob	8	SecurID Proteus
3	SecurID Watch	9	SecurID USBCOSMO (SID800)
4	SecurID software token	10	SecurID Flextoken

MinTokenLife is used to guarantee that a token has a minimum number of days before it expires. It is ignored if **MiscVariable** is set to -1.

SoftIDParams is **required** if SDTID files are used for token seed delivery, and n**ot required** when using CT-KIP. It must contain three decimal digits to control seed file generation characteristics:

First digit:

0 - Required but ignored.

Second digit - Copy Protection Flag:

0 - Copy protection off.

1 - Copy protection on.

Third digit - Password usage and Interpretation method:

0 - No password.

1 - Static password. See **SoftIDPW** below.

2 - Default login.

3 - Default login appended to static password.

SoftIDPW specifies a password. To be used **only** when **SoftIDParams** specifies a static password.

If **SoftTokenProfile** is not specified, a User is added and associated with a Token, but the Token is not deployed.

Nickname, **DeviceSerialNumber**, **SiteFile**, **SiteURL1**, **SiteURL2**, and **SiteURL3** can be used to assign values to software token device type attributes. For more information, see Software Token Device Type Attributes.

If the **--ctkip** command-line option is declared and **NicknameIsCtkipCode** is set to **true**, the contents of the **Nickname** field are used for the CT-KIP activation code. If **DeviceserialIsCtkipCode** is set to **true**, the contents of the **DeviceSerial** field are used for the CT-KIP activation code. Setting both options to **true** throws an error.

DeliveryMethod, **DestinationAddress**, **TemplateFile**, and **InstanceName** are used for automatic email notification during CT-KIP and software token database file provisioning. For more information, see Automatic Notification.

Add User to Group

Action	**AUG**
Required Fields	DefLogin or TokSerial, GrpName
Optional Fields	GrpDefLogin, GrpDefShell, IdentitySource, SecurityDomain, ForceGroupSearch

The User associated with the specified Token or **DefLogin** is added to the Group. The default User login name and Shell within this Group are assigned, unless the optional fields are used to change them.

Both the User and the Group must exist. The Group is not required to be present under the same domain as the User, but should be under the same realm.

If **ForceGroupSearch** is set **true**, the search for the Group name is carried out in the Internal Database, otherwise the search is carried out in the User's **IdentitySource**.

If **IdentitySource** and **SecurityDomain** are not provided, then the search for the User is carried out in the default realm.

Add Group

Action	**AG**
Required Fields	GrpName
Optional Fields	IdentitySource, SecurityDomain

The Group specified in the GrpName field is added.

If **IdentitySource** and **SecurityDomain** are not provided, then the Group is added to the default realm.

Add Group to Client

Action	**AGC**
Required Fields	GrpName, ClntName
Optional Fields	IdentitySource, SecurityDomain

Enables a Group of Users on a Client so that all members of the Group can authenticate on that Client. The Group and Client must exist.

If **IdentitySource** and **SecurityDomain** are not provided, then the search for the Group is carried out in the default realm.

Assign Profile

Action	**AP**
Required Fields	DefLogin or TokSerial, ProfileName
Optional Fields	IdentitySource, SecurityDomain

Assigns a profile specified by **ProfileName** to a User specified by **DefLogin** or **TokSerial.**

- If **DefLogin** is provided, and **IdentitySource** and **SecurityDomain** are not provided, the search for the User is carried out in the default realm.
- If **TokSerial** is provided, then **IdentitySource** and **SecurityDomain** are not required and are ignored, if present.

Add Security Domain

Action	**ASD**
Required Fields	SecurityDomainName, ParentDomainName
Optional Fields	SecurityDomainDescription, SecurityDomainCreatedBy, PolicyType1, PolicyName1, PolicyType2, PolicyName2, PolicyType3, PolicyName3, PolicyType4, PolicyName4, PolicyType5, PolicyName5

Adds a new Security Domain to the system.

Each **PolicyType** must have a matching **PolicyName** entry. Entries are case-sensitive.

PolicyName1 thru **5** specify the policy name to be selected for the matching **PolicyType**.

PolicyType1 thru **5** specify the policy type for policy selection.

Valid **PolicyType** values are:

- PasswordPolicy
- LockoutPolicy
- SelfServicePolicy
- AM_Token_Policy
- AM_OFFLINE_AUTHN_POLICY

Add Agent Host

Action	**AAH**
Required Fields	*Operation*
Optional Fields	AgentHostname, AgentHostAddress, AgentHostType, AgentNewHostname, AgentRestriction, EnableFlag, SecurityDomain, ClearNodeSecret, AgentTrustRealm

Performs the following operations for the Agent Host:

Add	The **AgentHost** specified in either the **AgentHostName** or **AgentHostAddress** field is added. The default **EnableFlag** value is true.
Update	Updates the attributes associated with the agent host specified by **AgentHostName**. To change the agent hostname, specify the **AgentHostName** and the **AgentNewHostName** attributes.
Remove	Removes the AgentHost, specified in the AgentHostName field.
List	Produces a list of **AgentHost** information for each **AgentHost**. The **AgentHost** selection can be filtered with three input fields **AgentHostType**, EnableFlag, **AgentRestriction**. If not specified, all **AgentHosts** are listed. The requested information is written to the Results File.

Add RADIUS Client

Action	**ARC**
Required Fields	CltnName, ClientIPAddress, RadiusClientModel, *Operation*, SharedSecret
Optional Fields	AnyRadiusClient, ClientIPAddressType, CreateAssociatedAgent, GrpName

The ARC (Add RADIUS Client) command adds a RADIUS client to the RSA Authentication Manager internal database. These RADIUS clients use IPv4 addresses. To add an IPv6 RADIUS client, you must use the Security Console, which allows you to add both IPv4 and IPv6 RADIUS clients. The ARC command supports the Add *Operation*.

You must add a RADIUS client to the deployment for each RADIUS device that is configured to use RSA SecurID as its authentication method, for example, a RADIUS-enabled device at the network perimeter, such as a VPN firewall server. The RADIUS client sends authentication requests to the RSA RADIUS server, which then forwards the request to RSA Authentication Manager.

You can configure RADIUS clients with or without an assigned authentication agent. The difference between the two methods is in the level of access control and logging you want to have.

- RADIUS client with an agent. Adding an agent to a RADIUS client allows Authentication Manager to determine which RADIUS client is used for authentication and to save this information in log files.
- RADIUS client without an agent. Without an assigned RADIUS client agent, Authentication Manager cannot track which RADIUS client sends authentication requests and you cannot assign a profile to the client.

You can use the same AMBA command for all of the following examples:

```
./rsautil AMBulkAdmin -a admin -P Password$ --verbose -m 0 -i
arc.csv
```

Where `Password$` is the password for the Super Admin.

Add a RADIUS Client Example

To add a RADIUS client named `WIN-9K35LDRF4P2.example.com` to the group rad in the Authentication Manager internal database, you can create a sample CSV file, called **arc.csv**, that contains the following data:

```
Action,ClntName,AnyRadiusClient,ClientIPAddressType,ClientIPAdd
ress,RadiusClientModel,SharedSecret,CreateAssociatedAgent,GrpNa
me,Operation
ARC, WIN-9K35LDRF4P2.example.com,N,0, 192.0.2.255,Cisco PIX
Firewall,1111,Y,rad,ADD
```

Add the Any RADIUS Client

You add an ANY RADIUS client if you do not want to track which RADIUS client sends authentication requests, for example, because you want to quickly add many RADIUS clients. Authentication requests using the shared secret specified for the ANY client are processed regardless of the originating client's IP address. An IP address is not added for the ANY client.

To add an ANY RADIUS client to the group radius in the RSA Authentication Manager internal database you can create a sample CSV file, called **arc.csv**, that contains the following data:

```
Action,ClntName,AnyRadiusClient,ClientIPAddressType,ClientIPAdd
ress,RadiusClientModel,SharedSecret,CreateAssociatedAgent,GrpNa
me,Operation
ARC,<ANY>,Y,0,,Aventail,2222,N,radius,ADD
```

Add RADIUS Clients in Bulk

You can add multiple RADIUS clients by adding more data to the CSV file.

To add five RADIUS clients, each with a restricted agent that is assigned to the group radius in the Authentication Manager internal database, you can create a sample CSV file, called **arc.csv**, that contains the following data:

```
Action,ClntName,AnyRadiusClient,ClientIPAddressType,ClientIPAdd
ress,RadiusClientModel,SharedSecret,CreateAssociatedAgent,GrpNa
me,Operation
ARC,new.example.com,N,0,192.0.2.91,Cisco ASA
Firewall,5555333,Y,radius,ADD
ARC,new.example.com,N,0,192.0.2.92,IP3
Networks,4333,Y,radius,ADD
ARC,new.example.com,N,0,192.0.2.93,Extreme
Networks,65333,Y,radius,ADD
ARC,new.example.com,N,0,192.0.2.94,Cisco PIX
Firewall,5555,Y,radius,ADD
ARC,new.example.com,N,0,192.0.2.95,Cisco ASA
Firewall,3333,Y,radius,ADD
```

Add Remote Group Client

Action	**ARGC**
Required Fields	GrpName, ClntName
Optional Fields	SecurityDomain

The ARGC (Add Remote Group Client) command links an authentication agent to a trusted group in the RSA Authentication Manager internal database.

If Security Domain is not provided, then the remote group is added in the default security domain (System Domain) created during Authentication Manager installation.

A trusted user group restricts access to an authentication agent that is enabled for trusted realm authentication. When you create a trusted user group, only members of the trusted user group can access the agent that is enabled for trusted realm authentication.

You can select which authentication agents that you want a trusted user group to have permission to access.

For example, a sample CSV file, called **argc.csv**, can contain the following data:

```
Action,GrpName,ClntName,SecurityDomain
argc,tgroup1,primary-PR-84.corp.emc.com
```

After you run the following AMBA command, the authentication agent listed above is added to the trusted user group in the RSA Authentication Manager internal database:

```
./rsautil AMBulkAdmin -a admin -P Password$ --verbose -m 0 -i
argc.csv
```

Single Softtoken Deployment

Action	**SSD**
Required Fields	TokSerial, TokEnabled
Optional Fields	FileName, SetPin, PinMode, PinType, SoftIDParams, SoftIDPW, IdentitySource, SecurityDomain, SoftTokenProfile, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, RegenerateSeed, NicknameIsCtkipCode, DeviceserialIsCtkipCode, DeliveryMethod, DestinationAddress, TemplateFile, InstanceName

Deploys a single software token either in a Software Token Database File or through CT-KIP. This action creates an unwrapped software token database file.

Prior to calling this function, the software token must be assigned to a User and the User must exist in the Authentication Manager internal database. The search for the Token is carried out in the given security domain, **SecurityDomain**.

SoftIDParams is **required** if SDTID files are used for token seed delivery, and n**ot required** when using CT-KIP. It must contain three decimal digits to control seed file generation characteristics:

First digit:

0 - Required but ignored.

Second digit - Copy Protection Flag:

0 - Copy protection off.

1 - Copy protection on.

Third digit - Password usage and Interpretation method:

0 - No password.

1 - Static password. See **SoftIDPW** below.

2 - Default login.

3 - Default login appended to static password.

SoftIDPW specifies a password. To be used **only** when **SoftIDParams** specifies a static password.

FileName is used to rename the SDTID file.

SetPin is used to assign a literal value as the PIN of the newly assigned Token.

IdentitySource is used to validate the security domain provided.

Nickname, **DeviceSerialNumber**, **SiteFile**, **SiteURL1**, **SiteURL2**, and **SiteURL3** can be used to assign values to software token device type attributes. For more information, see Software Token Device Type Attributes.

The **DeliveryMethod**, **DestinationAddress**, **TemplateFile**, and **InstanceName** fields are used for automatic email notification during CT-KIP and SDTID file provisioning. For more information, see Automatic Notification.

Note: Multiple Softtoken Deployment can be used to deploy a single token. It always creates an SDTID file wrapped in a zip file.

5 Change Actions

This chapter provides a description of the Change actions, and details the required and optional fields. Where fields are not described, see the description provided in the Input Field Definitions.

The topics in this chapter include:

Change or Add User

Action	**CAU**
Required Fields	DefLogin
Optional Fields	LastName, FirstName, Email, CertDN, DefShell, GrpName, GrpDefLogin, GrpDefShell, UserPwd, IdentitySource, SecurityDomain, EnableFlag, ForceGroupSearch, RemoveGrpDefLogin

Changes or adds the specified fields.

- If the **DefLogin** is not present in RSA Authentication Manager (Authentication Manager), a new account is created with a call to the **Add User** action, in which case the **LastName** field and a User password are also required.

  Provide each User with an individual password using the **UserPwd** optional field, or use the command-line option **--userpwd** to give the same password to all the Users. If the identity source is **Internal Database** and no password is provided, Users are added with a null password. If **IdentitySource** is an external directory like an LDAP server, the action throws an exception.

- If the User account exists, the User data fields **LastName**, **FirstName** and **DefShell** are modified if they differ from those in the database.

  Leaving the **UserPwd** field blank has no effect on the User's IMS password. To change the IMS password for a particular User, provide a value in the **UserPwd** field. Check Command-line Options and Input Field Definitions sections for more details.

To delete the **FirstName** or **DefShell**, supply a set of empty double quotes ("") in the input field. **LastName** cannot be deleted.

If a **GrpDefLogin** is supplied, the User is deleted from the Group and added back in as a member of the Group with the supplied **GrpDefShell** or its default. However, if the User is a member of the Group under the **DefLogin**, this Group membership is retained. Empty double quotes do not apply to the Group.

If **IdentitySource** and **SecurityDomain** are not provided, the search for the User is carried out in the default realm. Similarly, the User is added to the default realm. A User can be added only to a Group in the same **IdentitySource** or the Internal Database.

If **ForceGroupSearch** is set to **true**, the search for the Group name is carried out in the Internal Database. For any other value, including empty or missing, the search is carried out in the User's **IdentitySource**.

The **RemoveGrpDefLogin** option removes a logon alias from a group.
A logon alias allows users to log on with a user group ID.

Instead of removing a logon alias from a group, you can use AMBA to remove a user from a group with the **DUG** (Delete User from Group) command, but this command does not apply to external identity sources. For example, Authentication Manager cannot remove an Active Directory user from an Active Directory group. Instead, you can use the **RemoveGrpDefLogin** option for the **CAU** command to remove a logon alias from any group.

Change or Add User and Token

Action	**CAUT**
Required Fields	TokSerial
Optional Fields	DefLogin, FirstName, LastName, Email, CertDN, DefShell, TokEnabled, SetPin, PinMode, PinType, GrpName, GrpDefLogin, GrpDefShell, UserPwd, IdentitySource, SecurityDomain, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, NicknameIsCtkipCode, DeviceserialIsCtkipCode, EnableFlag, ForceGroupSearch, DeliveryMethod, DestinationAddress, TemplateFile

Changes the specified fields, however if the **TokSerial** is not assigned to a User account in the Authentication Manager Database, a new account is created with a call to the Add User and Token action.

If the **TokSerial** is associated with a User account, the User data fields **DefLogin**, **LastName**, **FirstName** and **DefShell**, are modified if they are different from those in the database.

To delete a **FirstName** or **DefShell**, supply a set of empty double quotes (“”) in the input field. **DefLogin** and **LastName** cannot be deleted.

TokEnabled and **SetPin** fields are updated if they are present and different from those found in the Token record.

If **GrpDefLogin** and **GrpName** are specified, the User is deleted from the Group and added back in as a member of the Group with the supplied **GrpDefShell,** or its default. However, if the User is a member of the Group under the **DefLogin,** or any other login, this Group membership is retained. Empty double quotes are not applicable to the Group.

If **IdentitySource** and **SecurityDomain** are not specified, the search for the User is carried out in the default realm. Similarly, the User is added to the default realm.

The **Nickname**, **DeviceSerialNumber**, **SiteFile**, **SiteURL1**, **SiteURL2**, and **SiteURL3** fields are used to assign values to software token device type attributes. For more information, see Software Token Device Type Attributes.

If **ForceGroupSearch** is set **true**, the search for the **GrpName** is carried out in the Internal Database. Any other value, including empty or missing, causes the search for the Group to be carried out in the User’s identity source.

The **DeliveryMethod**, **DestinationAddress**, and **TemplateFile** fields are used for automatic email notification during CT-KIP file provisioning For more information, see Automatic Notification.

Note: This action has been added as a convenience to some Users. It is recommended that extreme caution be used in selecting this action, as unintentional results are possible.

Change or Add User and Password

Action	**CAUP**
Required Fields	DefLogin
Optional Fields	LastName, FirstName, Email, CertDN, DefShell, GrpName, GrpDefLogin, GrpDefShell, UserPwd, IdentitySource, SecurityDomain, EnableFlag, SetPin, ForceGroupSearch

Changes the specified User data fields. Although **DefLogin** is the only required field, at least one of the optional fields should be specified.

- If the **DefLogin** exists in the database, the optional fields change an existing field. For example, **LastName**, **FirstName** and **DefShell**, are modified if they differ from those in the database.
- If the **DefLogin** does not exist, the **AUP** action is called, in which case the additional fields are required for a successful addition to occur.

  A User Password is required to add a User to the IMS layer. The IMS layer is RSA low level software that provides a database layer for multiple products. Either each User can be provided with different IMS password using the **UserPwd** optional field or the command-line option **--userpwd** can be used to set the same password for all the Users. If the Identity Source is Internal Database and if none is provided in both input file and as a command-line option, Users are added with an IMS password as null. If the Identity Source is an External Directory like an LDAP, the action throws an Exception. After an account is created, the IMS password is very rarely used for Authentication Manager functions and it is no way related to the Authentication Manager static password tokencode if one is declared.

To delete the **FirstName** or **DefShell**, supply a set of empty double quotes (“”) in the input field. **LastName** can not be deleted.

If **GrpName** and **GrpDefLogin** are specified, the User is deleted from the Group and added back in as a member of the Group with the supplied **GrpDefShell,** or its default. However, if the User is a member of the Group under the **DefLogin,** or any other login, this Group membership is retained. Empty double quotes are not applicable to the Group.

If **IdentitySource** and **SecurityDomain** are not specified, the search for the User is carried out in the default realm. Similarly, the User is added to the default realm.

SetPin specifies the User’s initial IMS password. The **SetPin** value becomes the User’s static passcode. It should not be confused with the IMS password defined in the following note.

To disable a static passcode, set the **SetPin** field to **disable**. For consistency, the **AUP** action will not allow **disable** as a static password.

If **ForceGroupSearch** is set **true**, the search for the **GrpName** is carried out in the Internal Database. Any other value, including empty or missing, causes the search for the Group to be carried out in the User’s Identity Source.

Change Principal Attribute Data

Action	**CPAD**
Required Fields	DefLogin, AttributeName
Optional Fields	SubCommand, AttributeValue, AttributeValue2, SecurityDomain, IdentitySource

Modifies existing custom attribute data for a custom attribute, identified by **AttributeName,** assigned to a User identified by **DefLogin**. The attributes can be either single-valued or multi-valued and can be of type Date, Integer, Float, Boolean, or String. Date, Integer, Float, and String types can be either single or multi-valued. Boolean attributes can only be single-valued.

Note: This action does not create or define new attributes. Currently there is no implementation for that in the AMBA utility. New attributes must be defined through the Security Console.

SubCommand identifies the specific operation to be performed on the attribute:

- values for Single-valued attributes:
  S | s | empty Set attribute to **AttributeValue**
- values for Multi-valued attributes:
  A | a Add **AttributeValue** to end of list of attribute values
  U | u Update/(replace **AttributeValue** with **AttributeValue2**
  R | r Remove **AttributeValue** from list of attribute values
  E | e Erase all values from attribute
  I | i Initialize attribute to **AttributeValue**.

Note: If a multi-valued attribute is empty either from an erase operation or because it is a newly created attribute that has never been populated, use the Initialize (I) sub-command to place the first value in the attribute list.

Failure to initialize a multi-valued attribute will result in the following error message if any other sub-command is used:

Failed to find target attribute: *attributeName*

After a multi-valued attribute has been initialized any of the other sub-commands can be used.

AttributeValue identifies the data to be added, replaced or removed.

AttributeValue2 identifies the data to replace **AttributeValue** for the **SubCommand**.

If **SecurityDomain** and **IdentitySource** are not specified, the search for the User is carried out in the default realm.

Change Principal Attribute Data Clear

Action	**CPADC**
Required Fields	DefLogin, AttributeName
Optional Fields	SecurityDomain, IdentitySource

Clears existing custom attribute data for a custom attribute, identified by **AttributeName**, assigned to a User identified by **DefLogin**. The attributes must be single-valued and can be of type Date, Integer, Float, Boolean, or String.

If **SecurityDomain** and **IdentitySource** are not specified, the search for the principal is carried out in the default realm.

Change User/Token Security Domain

Action	**CUSD**
Required Fields	DefLogin, DestinationSecurityDomain
Optional Fields	MiscVariable, IdentitySource, SecurityDomain, SubDomain, Limit

Changes the security domain of Users and/or Tokens to a new security domain. This action can changes the security domain of all Users or individual Users identified by default login. Optionally, it can change the security domain of all Tokens, all assigned or all unassigned Tokens.

DefLogin contains either a specific userID to identify a specific User account, or the value *all*, to identify all User accounts. Case is not significant for *all*.

DestinationSecurityDomain is used declare the new security domain. It must already exist in the Authentication Manager database.

MiscVariable is used to control whether or not Users only, Users and Tokens, or Tokens only are affected.

DefLogin	MiscVariable	Users Moved	Tokens Moved
Specific userID	0	Specified User	Assigned to Specified User[1]
	1	Specified User	None
all	0	All Users	All Tokens assigned to Users[1]
	1	All Users	None
	2	All Users	All Tokens
	3	None	Unassigned Tokens only
	4	None	All Tokens assigned to any Users[2]
	5	None	All Tokens [2]

[1] Tokens assigned to the User being moved are moved regardless of the Token's current security domain assignment. The determining factor is that the Token is "assigned" to a User that is a member of the source security domain.

[2] Tokens assigned to any User are moved without regard to the User's current security domain assignment. The determining factor is that the Token is a member of the source security domain.

If **IdentitySource** and **SecurityDomain** are not specified, the search for the User is carried out in the default realm.

The **SubDomain** and **Limit** fields apply only when **DefLogin** is set to *all*, otherwise they are ignored.

The default value for **SubDomain** in this action is 0 - do not include sub domains.

Change User Remote

Action	**CUR**
Required Fields	DefLogin, RemoteAlias, RealmName
Optional Fields	IdentitySource, SecurityDomain

Changes an existing local User to a remote User, or modifies an existing remote User to update the remote alias and realm information.

For an existing User, this function modifies the User record to indicate that they should be authenticated in the specified remote realm using the specified login name, **RemoteAlias**.

Note: There is no contact with the Authentication Manager instance in the remote realm. The changes are made locally. The login name in the remote realm, **RemoteAlias**, is not verified.

If **SecurityDomain** is not specified, the search for the User is carried out in the default security domain, System Domain.

If **SecurityDomain** is specified and is mapped to a different realm other than the default, **IdentitySource** must be provided.

Change PIN Status

Action	**CPS**
Required Fields	TokSerial
Optional Fields	SetPin, PinMode

Clears the Token's PIN, forces the PIN status to New PIN Mode, or sets the PIN to an explicit value.

PIN attributes, such as minimum or maximum length, alpha and/or numeric, are controlled by administrative settings. If you explicitly set a PIN, the Invalid PIN error message can result if one or more of these attributes is violated. The error message will not inform you as to the specific violation. In this case it is necessary to check with an Authentication Manager administrator to find out what attributes have been set in your system.

The following values are valid for **SetPin**:

C, c or 0	Clear the Token's PIN. Automatically forces the PIN status to New PIN Mode. **PinMode** is ignored.
Empty	With **PinMode** set to 1, forces the Token into New PIN Mode.
N	Sets the PIN status to New PIN Mode. The old PIN is required.
All other values	Sets the PIN to the supplied value.

Set **PinMode** to 1 to force the PIN status to New PIN Mode.

Change Token Status

Action	**CTS**
Required Fields	TokSerial, TokEnabled
Optional Fields	None

The specified assigned Token is enabled or disabled according to the value of the **TokEnabled** field.

Change Token Status eXtended

Action	**CTSX**
Required Fields	DefLogin or TokSerial, TokEnabled
Optional Fields	MiscVariable, SetPin, IdentitySource, SecurityDomain

Changes the enabled or disabled status of Tokens assigned to a specific User.

This action locates either all Tokens assigned to the User specified by **DefLogin**, or a specific Token, specified by **TokSerial**.

If **DefLogin** is provided and **IdentitySource** and **SecurityDomain** are not provided, then the search for the User is carried out in the default realm.

If **TokSerial** is provided, then the **IdentitySource** and **SecurityDomain** values are not required and are ignored, if present.

The **MiscVariable** field is used to specify the entities to update.

0 All Tokens and passwords (default).

1 Tokens only, no passwords

2 Passwords only, no Tokens

SetPin is used to enable a static password. If no **SetPin** value is provided, Static Password is enabled even if **TokEnabled** is set to 1.

Change Token Immediately

Action	**CTI**
Required Fields	TokSerial, ReplTokSerial
Optional Fields	None

Immediately replaces the Token specified by **TokSerial**, if the replacement Token, specified by **ReplTokSerial**, is currently unassigned.

Change Token on First Use of New Token

Action	**CTD**
Required Fields	TokSerial, ReplTokSerial
Optional Fields	None

Replaces the Token specified by **TokSerial** with a currently unassigned Token, specified by **ReplTokSerial**.

The first time the User uses the replacement Token the replaced Token is disabled. The User's PIN is preserved.

This action is being depreciated and is removed in a future release. Use the **REPT** or **REPTA** action instead. For more information, see Replace Token or Replace Token Automatic.

Change Temporary User Mode

Action	**CTU**
Required Fields	DefLogin or TokSerial
Optional Fields	DefShell, ClntDefLogin, GrpDefShell, GrpDefLogin, IdentitySource, SecurityDomain

Puts the User identified by login or Token, in temporary mode for a specified duration, or removes the User from temporary mode.

If **DefLogin** is provided and **IdentitySource** and **SecurityDomain** are not provided, then the search for the User is carried out in the default realm.

If **TokSerial** is provided, then the **IdentitySource** and **SecurityDomain** values are not required and are ignored, if present.

Both start and end times or end time only can be set, in which case temporary mode lasts from now until the end time. Both date and hour must be specified for start and end times.

To remove the User from temporary status, leave all optional fields empty or specify 00/00/0000 in the **DefShell** field.

The optional fields have non-standard usage and are used to provide the following values:

DefShell	End Date	empty | mm/dd/yyyy | 00/00/0000)
ClntDefLogin	End Time	empty | h | hh) where 0 >= hours <= 23
GrpDefShell	Start Date	empty | mm/dd/yyyy)
GrpDefLogin	Start Time	empty | h | hh) where 0 >= hours <= 23

Set Emergency Access Fixed

Action	**EAFXD**
Required Fields	TokSerial, SetPin
Optional Fields	CreatePin

Sets the status of the specified Token to Fixed and assigns a fixed password. The lifetime of the password can be defined in local time by using either the dateExpire, hourExpire, or lifeTime arguments.

SetPin supplies the password. Its acceptability is governed by system parameters.

Set Emergency Access OTP

Action	**EAOTP**
Required Fields	TokSerial
Optional Fields	SetPin, CreatePin

Sets the status of the specified Token to lost and generates a set of one-time passwords (OTPs) for the Token. By default, this function returns a set of two OTPs. You can specify a larger number of passwords. These are given to the User and can be used for authentication. The lifetime of the OTP can be defined in local time by using either the dateExpire, hourExpire, or lifeTime arguments.

SetPin is used to specify the number of passwords to generate. The format is *Nn* where *n* represents the number of OTPs to generate, to a maximum of 50. Length and format of OTPs depend on Token policy.

Set Emergency Access OFF

Action	**EAOFF**
Required Fields	TokSerial
Optional Fields	None

Switches off emergency access mode for the specified Token. The User's one-time password(s) is/are destroyed and the status of the original Token is changed from Lost to Enabled.

Extend Software Token Lifetime

Action	**ETL**
Required Fields	TokSerial
Optional Fields	UseSameSecurityDomain

Extends the lifetime of a distributed Token that has or is about to expire. Authentication Manager selects an unassigned Token that has the longest lifetime, assigns its expiration date to the original Token, and then deletes the Token that provided its expiration date.

By extending Token lifetimes, you can avoid replacing expired Tokens on User devices, such as mobile phones, tablets, and PCs. Token provisioning is required to occur only once on each User device, and Authentication Manager assumes full administrative control over whether an extended Token is available for authentication.

The optional **UseSameSecurityDomain** field requires AMBA to only select unassigned software tokens that share the same security domain as the tokens that are being extended.

Set Software Token Profile

Action	**SSTP**
Required Fields	SoftTokenProfile
Optional Fields	None

Defines a global software token profile.

A software token profile is required to provision software tokens with CT-KIP and Software Token Database File files. The software token profile name in the AMBA utility must match the name defined in the Security Console.

This action applies to actions that provision individual software tokens, such as **ATU**, **ATUA**, **AUT**, **AUTA**, **REPT**, **REPTA**, **SSD**, and **CIF**. It does not apply to bulk and multiple software token operations, for example, **MSD**, **MTA**, or **MTR**.

For examples of the use of this action, see Software Token Distribution.

Update User Data

Action	**UUD**
Required Fields	DefLogin or TokSerial
Optional Fields	LastName, FirstName, Email, CertDN, DefShell, DefLogin, MiscVariable, UserPwd, IdentitySource, SecurityDomain, EnableFlag

This action updates User data. To delete a field that is not required, supply a set of double quotes, for example, "".

To modify the **DefLogin** field, supply the **TokSerial** field or enter the old **DefLogin** in the **MiscVariable** field. **TokSerial** takes precedence over **MiscVariable**. If the **TokSerial** field is not present and **MiscVariable** contains a value, AMBA attempts to obtain a **TokSerial** using this value. If a **TokSerial** is obtained, the **DefLogin** is changed.

If **DefLogin** is provided and **IdentitySource** and **SecurityDomain** are not provided, then the search for the User is carried out in the default realm.

If **TokSerial** is provided, then the **IdentitySource** and **SecurityDomain** values are not required and are ignored, if present.

Note: This action can also be used to update the IMS Password for a User. Either each User can be provided with different password using **UserPwd** optional field or the command-line option **--userpwd** can be used to have the same password given for all the Users. Leaving both the fields blank will cause no effect on existing Passwords.

User Admin Role

Action	**UAR**
Required Fields	DefLogin, *Operation*
Optional Fields	IdentitySource, SecurityDomain, RoleName, RoleSecurityDomain

This action updates the Administrative Roles associated with the User.

To assign an Administrative Role to the User, or to unassign an individual Administrative Role from the User, **RoleName** must be provided.

To unassign all the Roles which are assigned to the User, **DefLogin** and the valid *Operation* must be provided.

Supported values for *Operation*:

- Add Assigns the Administrative Role to the User.
- Remove Unassigns the Administrative Role from the User.
- Clear Unassigns all Administrative Roles for the User.

If **IdentitySource** and **SecurityDomain** are not provided, the search for the User is carried out in the default realm.

If the **RoleSecurityDomain** is not provided, the search for the Role is carried out in the default security domain.

Change Token Attributes

Action	**CTA**
Required Fields	TokSerial, *Operation*, AttributeName
Optional Fields	AttributeValue, DestinationAttributeName

This action updates the Token Attributes associated with the Token. The action can be used to add/update/clear Token Attributes associated with a Token.

To assign a Token Attribute to a Token, or to unassign a Token Attribute from a Token, **TokSerial**, the valid *Operation* and the **AttributeName** must be provided.

All fields must be present to update a Token attribute name.

Supported values for *Operation*:

- Add Assigns the Token Attribute to the Token.
- Clear Unassigns the Token Attribute associated with the Token.
- Update Modifies the Token Attribute Name and Value associated with the Token.

If the **AttributeValue** is not present, the default value associated with the Custom Token Attribute is used. If both **AttributeValue** and the default value are blank, an error message is thrown.

Change Token Security Domain

Action	**CTSD**
Required Fields	TokSerial, DestinationSecurityDomain
Optional Fields	EndRange, SecurityDomain, TokenType, TokenAssigned, SubDomain, Limit

This action is used to move a range of Tokens to a new security domain.

TokSerial specifies the serial number of the first Token in a range to be moved. If **EndRange** is not specified, only one Token is moved.

DestinationSecurityDomain must already exist in the Authentication Manager database.

TokenType specifies the type of RSA SecurID (SecurID) Token to be moved. If not specified all the Tokens are moved irrespective of type. Supported values are:

0	SecurID Standard Card	6	SecurID Modem
1	SecurID PINPAD Card	7	SecurID Crypto
2	SecurID Key Fob	8	SecurID Proteus
3	SecurID Watch	9	SecurID USBCOSMO (SID800)
4	SecurID software token	10	SecurID Flextoken
5	SecurID Smartcard		

The default value for **SubDomain** in this action is 0 - do not include sub domains.

Change Date Format

Action	**CDF**
Required Fields	None
Optional Fields	**MiscVariable**

This action is used to reset or format Java Date objects for the List actions **LUIF**, **LUI**, **LTIF** and **LTI**. It is used to override the default date format which is normally empty. This action will replace any date format created with the **datefmt** command-line and INI file options. The action can be used anywhere in the input file and as many times as required.

MiscVariable is used to provide a Java **SimpleDateFormat** string to use. If the field is absent or empty the date format is cleared and returned to its default state. No syntax checking is performed on this value so errors can occur when the format is first used and incorrect results can be encountered.

Date Format Examples:

yyyy/MM/dd

yyyy-MM-DD HH:mm:ss

Do not use quotes in these strings unless you want them to appear in the output.

The following pattern letters are defined. Letter case is significant. All alphabetic characters not defined, both upper and lower case, are reserved:

Table 17: Defined Pattern Letters

Letter	Date or Time Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24

Table 17: Defined Pattern Letters (Continued)

Letter	Date or Time Component	Presentation	Examples
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800

Change Input Format

Action	**CIF**
Required Field Names	Action
Optional Field Names	IdentitySource, SecurityDomain, LastName, FirstName, DefLogin, DefShell, UserPwd, TokSerial, ReplTokSerial, TokEnabled, SetPin, CreatePin, PinMode, PinType, GrpName, GrpDefLogin, GrpDefShell, ClntName, AgentHostName, AgentNewHostname, AgentTrustRealm, SoftIDParams, SoftIDPW, RemoteAlias, RealmName, CompareField, CompareType, CompareValue, OutputOption, ExtnDataOption, MiscVariable, ProfileName, TokenSerial, RangeMode, StartRange, EndRange, Password, FileName, CopyProtect, OverOption, Email, CertDN, Key, KeyType, SoftTokenProfile, SecurityDomainName, ParentDomainName, SecurityDomainDescription, SecurityDomainCreatedBy, PolicyType1, PolicyName1, PolicyType2, PolicyName2, PolicyType3, PolicyName3, PolicyType4, PolicyName4, PolicyType5, PolicyName5, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, NicknameIsCtkipCode, DeviceserialIsCtkipCode, DestinationSecurityDomain, EnableFlag, ForceGroupSearch

The **CIF** action is used to override the default input file format and simplify the input file build process by allowing the User to define command-lines unique to each User's requirement. It can be used anywhere in the input file and as many times as desired. It is also used to dynamically redefine the input file format at run time.

The **CIF** action is case insensitive. The number and order of field names is arbitrary.

A **CIF** action is simply a header preceded by the **CIF** action. It must always contain an **Action** field. Any additional parameters must be picked from the available field names listed above.

The **Action** field and any other fields can be arranged in any order. Although not required, it is good programming practice to make the **Action** field the first field of all input lines. If the **Action** field is positioned in other than the first field, then a subsequent **CIF** action must be carefully formatted to ensure that the **Action** appears in the correct column. An example of a **CIF** action is:

CIF,Action,GrpName

This implies that all input following this action consists only of an action code followed by a **GrpName**. This example is useful where a large number of Groups are to be created. To create each Group, provide the action, such as **AG** and the Group name.

In the following example, the first set of input is defined as add Group, followed by add User to Group, followed by add User with Token:

```
CIF,Action,DefGrpname
AG,group1
AG,group2
AG,group3
.
.
.
CIF,Action,DefLogin,GrpName
AUG,user1,group1
AUG,user2,group1
AUG,user3,group2
AUG,user4,group2
.
.
.
CIF,Action,LastName,DefLogin,TokSerial,TokEnabled
AUT,LN1,l1,11223344,1
AUT,ln2,l2,33445677,1
.
.
.
```

Note: If a **CIF** action fails for any reason, all input lines are logged and ignored until a valid **CIF** action is processed or the end of file is reached, which ever occurs first.

Change the Results File Name

Action	**CRFN**
Required Fields	None
Optional Fields	**MiscVariable**

This action is used to change the Results File name any number of times during an AMBA session. The results file is opened and closed at the command level, not the session level. This action can be used between multiple listing actions in one session. This will prevent one list action overwriting the results of a previous list action in any one session.

The result file name is supplied in the **MiscVariable** field. This variable can include a drive, path and extension, where applicable.

When **MiscVariable** is empty the Result File name is reset to the default value.

6 Delete Actions

This chapter provides a description of the Delete actions, and details the required and optional fields.

Where fields are not described, see the description provided in the Input Field Definitions.

The topics in this chapter include:

- Delete User from Group
- Delete Group from Client
- Delete Remote Group Client
- Delete Group
- Delete Remote Group
- Delete Security Domain
- Delete Token
- Delete User
- Delete User from Remote Group
- Unassign Profile
- Unregister External User
- Replace Token
- Replace Token Automatic
- Rescind Token
- Unassign Token.

Delete User from Group

Action	**DUG**
Required Fields	DefLogin, GrpName
Optional Fields	IdentitySource, SecurityDomain, ForceGroupSearch

Removes the User identified by the **DefLogin** field from the Group.

If **IdentitySource** and **SecurityDomain** are not provided, the search for the User is carried out in the default realm.

Delete Group from Client

Action	**DGC**
Required Fields	GrpName, ClntName
Optional Fields	IdentitySource, SecurityDomain

Removes the Group from the Client.

If **IdentitySource** and **SecurityDomain** are not provided, the search for the Group is carried out in the default realm.

Delete Remote Group Client

Action	**DRGC**
Required Fields	GrpName, ClntName
Optional Fields	SecurityDomain

The **DRGC** (Delete Remote Group Client) command removes an authentication agent from a trusted user group in the RSA Authentication Manager internal database.

If Security Domain is not provided, then the remote group is added in the default security domain (System Domain) created during Authentication Manager installation.

A trusted user group restricts access to an agent that is enabled for trusted realm authentication. When you create a trusted user group, only members of the trusted user group can access the agent that is enabled for trusted realm authentication.

You can unselect which authentication agents that you want a trusted user group to not have permission to access.

For example, a sample CSV file, called **drgc.csv**, can contain the following data:

```
Action,GrpName,ClntName,SecurityDomain
drgc,tgroup1,primary-PR-84.corp.emc.com
```

After you run the following AMBA command, the authentication agent listed above is removed from the trusted user group in the RSA Authentication Manager internal database:

```
./rsautil AMBulkAdmin -a admin -P Password$ --verbose -m 0 -i
drgc.csv
```

Delete Group

Action	**DG**
Required Fields	GrpName
Optional Fields	IdentitySource, SecurityDomain

Deletes the Group record from the database.

If **IdentitySource** and **SecurityDomain** are not provided, the search for the Group is carried out in the default realm.

Delete Remote Group

Action	**DRG**
Required Fields	GrpName
Optional Fields	SecurityDomain

The **DRG** (Delete Remote Group) command removes a remote user group from the RSA Authentication Manager internal database.

If Security Domain is not provided, then the remote group is added in the default security domain (System Domain) created during Authentication Manager installation.

Delete a trusted user group to revoke a user group's access privileges on a trusted realm.

For example, a sample CSV file, called **drg.csv**, can contain the following data:

```
Action,GrpName
drg, tgroup1
drg, tgroup2
drg, tgroup3
```

After you run the following AMBA command, the three user groups listed above are removed from the RSA Authentication Manager internal database:

```
./rsautil AMBulkAdmin -a admin -P Password$ --verbose -m 0 -i
drg.csv
```

Delete Security Domain

Action	**DSD**
Required Fields	SecurityDomainName, ParentDomainName
Optional Fields	None

Deletes the Security Domain from the system.

Delete Token

Action	**DT**
Required Fields	TokSerial
Optional Fields	None

Deletes the record of an unassigned Token from the database.

Delete User

Action	**DU**
Required Fields	DefLogin or TokSerial
Optional Fields	IdentitySource, SecurityDomain, SubDomain

Deletes the specified User from the database and returns any associated Tokens to the unassigned state. The User is removed from all other associations, such as Groups.

If **IdentitySource** and **SecurityDomain** are not provided, the search for the User is carried out in the default realm.

The default **SubDomain** for this action is 1, include sub domains.

Delete User from Remote Group

Action	**DURG**
Required Fields	GrpName, DefLogin
Optional Fields	SecurityDomain

The **DURG** (Delete User from Remote Group) command removes a trusted user from a trusted user group in the RSA Authentication Manager internal database.

If Security Domain is not provided, then the remote group is added in the default security domain (System Domain) created during Authentication Manager installation.

You can remove existing trusted users from an existing trusted user group.

For example, a sample CSV file, called **durg.csv**, can contain the following data:

```
Action, GrpName, DefLogin, SecurityDomain
durg,tgroup1,tuser1
```

After you run the following AMBA command, the trusted users listed above are removed from the trusted user group in the RSA Authentication Manager internal database:

```
./rsautil AMBulkAdmin -a admin -P Password$ --verbose -m 0 -i
durg.csv
```

Unassign Profile

Action	**UP**
Required Fields	DefLogin or TokSerial
Optional Fields	IdentitySource, SecurityDomain, SubDomain

Unassigns the currently assigned User profile.

IdentitySource and **SecurityDomain** set the location of the search for the User.

- If **DefLogin** is provided, **IdentitySource** and **SecurityDomain** values are used.
- If **TokSerial** is provided, **IdentitySource** and **SecurityDomain** values are not used and are ignored, if present.

The default **SubDomain** for this action is 1, include sub domains.

Unregister External User

Action	**UEU**
Required Fields	DefLogin
Optional Fields	IdentitySource, SecurityDomain

Unregisters a User in an external identity source.

Replace Token

Action	**REPT**
Required Fields	TokSerial, ReplTokSerial,
Optional Fields	TokEnabled, SetPin, PinMode, PinType, SoftIDParams, SoftIDPW, FileName, CopyProtect, IdentitySource, SecurityDomain, SiteFile, SiteURL1, SiteURL2, SiteURL3, DeviceSerialNumber, DeviceserialIsCtkipCode, SoftTokenProfile, Nickname, NicknameIsCtkipCode, RegenerateSeed, DeliveryMethod, DestinationAddress, TemplateFile

Assigns a Token to a User as a replacement Token. The Token to be replaced, specified by **TokSerial**, must already be assigned to a User. If the replacement Token, specified by **ReplTokSerial**, is already assigned, a **Failure** message is generated. The replacement Token is not required to be under the same domain as the User, but should be present in the same realm.

If software tokens are assigned, the **-g** and **-gdir** Command-line Options can be used to instruct the AMBA utility to generate Software Token Database Files and place them in the specified directory. If the **-g** or **--ctkip** options are not used, this action assigns Tokens but does not build any output files or generate any CT-KIP credentials. The Set Software Token Profile action can be used to force a specific device type for credential generation.

SoftIDParams is **required** if SDTID files are used for token seed delivery, and n**ot required** when using CT-KIP. It must contain three decimal digits to control seed file generation characteristics:

First digit:

0 - Required but ignored.

Second digit - Copy Protection Flag:

0 - Copy protection off.

1 - Copy protection on.

Third digit - Password usage and Interpretation method:

0 - No password.

1 - Static password. See **SoftIDPW** below.

2 - Default login.

3 - Default login appended to static password.

SoftIDPW specifies a password, to be used **only** when **SoftIDParams** specifies a static password.

The following fields are ignored if **--ctkip** is **true**:

- **SoftIDParams**
- **FileName**
- **CopyProtect**.

IdentitySource is used to validate the security domain provided. The search for the Users is carried out in the given identity source.

SecurityDomain is used for the search for the Tokens.

If a **SoftTokenProfile** attribute is not provided with **AUTA**, then a User is added and associated with a Token, but the Token is not deployed.

Nickname, **DeviceSerialNumber**, **SiteFile**, **SiteURL1**, **SiteURL2**, and **SiteURL3** assign values to Software Token Notifications and Attributes. A specific software token device type can be linked to a software token through the Security Console, the API, or the **SSTP** action.

The **DeliveryMethod**, **DestinationAddress**, **TemplateFile**, and **InstanceName** fields are used for automatic email notification during CT-KIP and SDTID file provisioning For more information, see Automatic Notification.

Replace Token Automatic

Action	**REPTA**
Required Fields	TokSerial, MiscVariable,
Optional Fields	TokEnabled, SetPin, PinMode, PinType, MinTokenLife, SoftIDParams, SoftIDPW, FileName, CopyProtect, IdentitySource, SecurityDomain, SiteFile, SiteURL1, SiteURL2, SiteURL3, DeviceSerialNumber, DeviceserialIsCtkipCode, SoftTokenProfile, Nickname, NicknameIsCtkipCode, RegenerateSeed, DeliveryMethod, DestinationAddress, TemplateFile

This action automates the Replace Token action by obtaining an unassigned Token of a specified type from the system and calling the **REPT** action using the newly acquired Token serial number.

This action is identical to the **REPT** action except a token type is specified instead of the replacement Token, and the AMBA utility attempts to find an unassigned Token of the requested type.

The field definitions and requirements are identical to those of the **REPT** action with the following exceptions:

The **MiscVariable** field is used to specify the token type for the replacement RSA SecurID (SecurID) Token. Acceptable values are:

-2	Same as Token being replaced	5	SecurID Smartcard
-1	First available unassigned Token	6	SecurID Modem
0	SecurID Standard Card	7	SecurID Crypto
1	SecurID PINPAD Card	8	SecurID Proteus
2	SecurID Key Fob	9	SecurID USBCOSMO (SID800)
3	SecurID Watch	10	SecurID Flextoken
4	SecurID software token		

MinTokenLife is used to guarantee that a replacement Token has a minimum number of days before it expires. It is ignored if **MiscVariable** is set to -1.

The search for the Token to be assigned is carried out in the same realm as that of the User while assigning a Token of a particular type. However if the first available unassigned Token is requested, **MiscVariable = -1**, then the RSA Authentication Manager API retrieves the Token which expires first, irrespective of Security Domain.

If successful, the newly acquired Token serial number is inserted in the **ReplTokSerial** field of the **REPT** action which is called. If that operation is successful, the serial number of the newly assigned Token is reported in the AMBA transaction log.

Rescind Token

Action	**RT**
Required Fields	TokSerial
Optional Fields	None

Unassigns the specified Token. No other action regarding this Token or the associated User is performed.

Unassign Token

Action	**UT**
Required Fields	TokSerial
Optional Fields	None

Unassigns a Token from the User.

CAUTION: If the User has no other Tokens, this function also deletes the User record from the database, provided that the following conditions apply:

- The User is not an administrator
- The User is not enabled on any Agent Host
- The User does not belong to any Group
- The User record has no extension fields.

Unless all of these requirements are met, the Token is not unassigned nor is the User deleted. To un-assign the Token and leave the User account in place, use Rescind Token.

7 On-Demand and Risk-Based Authentication Actions

This chapter describes the actions and input fields used for on-demand authentication (ODA) and risk-based authentication (RBA):

- ODA is a service that allows users to receive on-demand tokencodes delivered by text message (SMS) or e-mail (SMTP). A tokencode is a randomly generated six- or eight-digit number. ODA can be used to protect resources, such as an SSL-VPN, thin client, or web portal.
- RBA identifies potentially risky or fraudulent authentication attempts by silently analyzing User behavior and the device of origin. RBA strengthens RSA SecurID authentication and traditional password-based authentication. If the assessed risk is unacceptable, the User is challenged to further confirm their identity using ODA or security questions that the User must correctly answer.

Where fields are not described, see the description provided in the Input Field Definitions.

ODA and RBA must be configured in RSA Authentication Manager (Authentication Manager). For more information, see the following documentation:

- "On-Demand Authentication" on RSA Link at **https://community.rsa.com/docs/DOC-76775**.
- "Risk-Based Authentication" on RSA Link at **https://community.rsa.com/docs/DOC-77387**.

Where fields are not described, see the description provided in the Input Field Definitions.

The topics in this chapter include:

Enable OnDemand Authentication

Action	**EODA**
Required Fields	DefLogin, PINIndicator, SetPin
Optional Fields	IdentitySource, SecurityDomain, InstanceName, PinMode, ExpiryDate, DeliveryMethod, DestinationAddress, TemplateFile, OutputOption

Enables an ODA for a User. A PIN can be assigned or system generated. Some field names are duplicates of field names used for Token management but can have slightly different usage. Consult this guide and the Authentication Manager Help for the correct definition of ODA action fields. Results are written to the Results File.

PINIndicator specifies the type of PIN and PIN mode assignment. Acceptable values or this action are:

SET_TEMP_PIN	the ODA is initialized with the provided PIN. Authenticator is in New PIN Mode.
SET_PERM_PIN	the ODA is initialized with the provided PIN. Authenticator is **not** put into New PIN Mode.
GENERATE_PIN	ODA is initialized with a system-generated PIN. Authenticator is **not** put into New PIN Mode. PIN format is determined by the User's token policy.
NO_PIN_UPDATE	the User's PIN is not modified.

The optional fields are ignored when **PINIndicator** is set to **GENERATE_PIN**.

SetPin specifies the initial PIN value when enabling, when **PINIndicator** is set to **SET_TEMP_PIN** or **SET_PERM_PIN**.

SetPin, PinMode and **DestinationAddress** are ignored when **PINIndicator** is set to **GENERATE_PIN**.

InstanceName is used to specify name of current instance. This is required only if Notify is set as an **OutputOption** and multiple Instances are declared.

OutputOption can be any combination of the following:.

A | a	Append output to an existing results file.
H | h	Write a header record to the results file.
N | n	Notify. Send an SMTP message with a generated PIN to principal. Message type depends on **DeliveryMethod**.
O | o	Omit the PIN from the results file. PIN is replaced by 'omitted'.

Update OnDemand Authentication

Action	**UODA**
Required Fields	DefLogin, PINIndicator, SetPin
Optional Fields	IdentitySource, SecurityDomain, InstanceName, PinMode, ExpiryDate, DeliveryMethod, DestinationAddress, TemplateFile, OutputOption

Updates existing User ODA settings. Some field names are duplicates of field names used for token management but can have slightly different usage. See the Authentication Manager Help for the correct definition of ODA action fields. Results are written to the Results File.

PINIndicator specifies the type of PIN update. Acceptable values or this action are:

SET_TEMP_PIN	the ODA is updated with the provided PIN. Authenticator is in New PIN Mode.
SET_PERM_PIN	the ODA is updated with the provided PIN. Authenticator is **not** put into New PIN Mode.
NO_PIN_UPDATE	the User's PIN is not modified.

SetPin specifies the initial PIN value when enabling ODA, when **PINIndicator** is set to **SET_TEMP_PIN** or **SET_PERM_PIN**.

SetPin, PinMode and **DestinationAddress** are ignored when **PINIndicator** is set to **NO_PIN_UPDATE**.

InstanceName is required only f Notify is set as an **OutputOption** and multiple Instances have been declared.

OutputOption can be any combination of the following:

A | a	Append output to an existing results file.
H | h	Write a header record to the results file.
N | n	Notify. Send an SMTP message with a generated PIN to principal. Message type depends on **DeliveryMethod**.
O | o	Omit the PIN from the results file. PIN is replaced by ‘omitted’.

Disable OnDemand Authentication

Action	**DODA**
Required Fields	DefLogin
Optional Fields	IdentitySource, SecurityDomain, OutputOption

Disables an ODA for a User, identified by DefLogin. The results are written to the Results File.

OutputOption can be any combination of the following:

A | a	Append output to an existing results file.
H | h	Write a header record to the results file.

Enable or Disable Risk-Based Authentication

Action	**RBA**
Required Fields	DefLogin, RBAOperation
Optional Fields	SecurityDomain, IdentitySource, SubDomain

Enables or disables Risk-Based Authentication (RBA) for the User specified in **DefLogin**, and can clear the User's security questions or device mappings.

RBAOperation specifies one of the following operations for each User:

Enable	Enables the User for RBA.
Disable	Disables the User for RBA.
ClearSQ	Clears the User's set of security questions.
ClearDevice	Clears the device mapping for the User.

For example:

```
Action,DefLogin,RBAOperation
RBA,RBAUser1,ClearSQ
RBA,RBAUserB,Enable
```

8 List Actions

The List actions produce data that can be used as is or passed on to some other product for additional sorting, formatting or combination with data from other sources. The output of the List actions can be anything from a simple list of User default logins or Token serial numbers to a full line of combined Token and User information.

The List actions extract data from the Ace server database based on input selection criteria. The input selection criteria can result in a list of everything from all Users or Tokens to a very narrow range of Users or Tokens. Additional data such as all User information, extended User information, User extension data and Group membership information can be reported.

Token information for assigned Tokens can be included. For Token-based reports, full Token information and Token extension data can be reported along with User and Group information if the Token is assigned to a User.

Most of the List reporting actions are available to the multiple Token assignment and multiple Token replacement actions. If these actions are used to assign or replace Tokens, the report actions can be used for Token control or notification.

The output of these actions is a comma separated variable list that can be processed by most scripting languages, spreadsheet programs or word processor programs. If the report is redirected to **STDOUT** or **STDERR**, the last line of output is the single word 'Done' followed by a newline sequence. In this case, a report that generates no output consists of a single line beginning with the word 'Done'.

Where fields are not described, see the description provided in the Input Field Definitions.

The topics in this chapter include:

List User Information by Field

Action	**LUIF**
Required Fields	None
Optional Fields	CompareField, CompareType, CompareValue, OutputOption, ExtnDataOption, MiscVariable, SecurityDomain, IdentitySource

Produces a list of default logins or a list of User information for each default login. The requested information is written to the Results File.

The default logins selected are based on the values in **CompareField**, **CompareType**, and **CompareValue**. The listing is controlled by **OutputOption ExtnDataOption** and **MiscVariable**.

Select the **CompareField** and the **CompareType** values from the following table and supply a value for **CompareValue** of the type listed in the value column.

Table 18: Values for User Listings

User Listed	Field	Type	Value
All	0	0	Ignored
By last name	1		
All beginning with		1	String
All matching		2	String
All containing string		3	String
All with any value		5	Ignored
By first name	2		
All beginning with		1	String
All matching		2	String
All containing string		3	String
All with any value		5	Ignored
By default login	3		
All beginning with		1	String
All matching		2	String
All containing string		3	String
All with any value		5	Ignored

Table 18: Values for User Listings (Continued)

User Listed	Field	Type	Value
Local or remote	5		
All local		1	Ignored
All remote[1]		2	Ignored
Permanent or Temporary	6		
All permanent		1	Ignored
All temporary		2	Ignored
By Tokens assigned	7		
All with specified number of Tokens		1	Number
All with at least one replacement pair		2	Ignored
All with passwords		3	Ignored
All with expired Tokens		4	Ignored
All with lost Tokens		5	Ignored
All with Token type		6	Token Type[2]
All with Tokens expiring in # of days		7	Number
By LDAP data[3]	8		
All beginning with		1	String
All matching		2	String
All containing string		3	String
All without a value (empty)		4	Ignored
All with any value		5	Ignored
By profile	9		
All with a named profile		3	String
By User extension	10		
All with extensions		1	Ignored
All without extensions		2	Ignored
All with extension keys		3	String
All without extension keys		4	String

[1]For Remote Users standard header in the results file would be
chLoginName, chTrustedrealmName, chSecurityDomain, chDefShell, chProfileName, chRemoteAlias, *tagBGM*,GrpName.
Other than Append Value(a) in Output option, every other data in ExtnDataOption and OutputOption is ignored.

[2]For Token type values, see the TokenType in Input Field Definitions:

[3]List User Info by LDAP data outputs the same data as List User Info by last name. This option is provided in order to keep the AMBA utility aligned as closely as possible with the ACEBulkAdmin.

OutputOption can contain the following values:
[[-2 | -1 | 0 - 7 | 30 - 34 | 60] [H | h] [A | a]]

where:

0 List User information for all selected default logins, do not write a header record (default).

60 option 30 plus two replacement fields for the assigned Tokens. The two replacement fields are appended to the Token info. The two fields are:

ReplacementMode

`NO_REPLACEMENT_TKN` - Token has no replacement
`HAS_REPLACEMENT_TKN` - Token has a replacement Token
`IS_REPLACEMENT_TKN` - Token is a replacement Token
`HAS_BEEN_REPLACED` - Token has been replaced by another Token.

ReplaceTknSN

Returns the serial number of the Token number that replaced or is replacing this Token.

If the Append mode is not specified, then new is assumed. Append and Header can be used simultaneously, which can result in header lines within the file body. This is intentional and it is up to the User to determine which options make sense and whether or not down stream applications are able to correctly process this file.

Because the **CIF** action can be used between List actions, if the Append option is used, data columns might not line up. This is intentional and the User must choose the options appropriate for the desired outcome.

For the definitions of the standard **OutputOption** values, refer to OutputOption Values.

ExtnDataOption can contain the following values:

0 - no additional data listed

1 - include Token extension data

2 - include User extension data

3 - option 1 and 2

4 - include Group membership Group names

5 - option 1 and 4

6 - option 2 and 4

7 - option 3 and 4

8 - include Group membership field values

9 - option 1 and 8

10 - option 2 and 8

11 - option 3 and 8

Because these options can appear zero or more times, formatting tags are inserted to assist in parsing the data. If User extension data appears in the output, the first User extension data key is preceded by *tagBUED*, followed by as many User extension key/data pairs as exist for this User. If there are no User extension key/data pairs, the tag will not appear. If Group membership data appears in the output, the first Group membership name (options 4 and 6) or the first Group membership default login, options 8 and 10, is preceded by the tag *tagBGM*. If there are no Group membership values, the tag will not appear.

If Token information is requested for Tokens assigned to a User, the Token information is appended after any of the above information. Each Token is listed as *tagBTOK*, Token info. If Token extension data is requested, it follows the applicable Token as *tagBTED*, followed by the Token extension data key/data pairs that exist for this Token.

If a header record is requested, appropriate field names are written for each option. The field names appear only once, even though the actual data can appear more than once. When appropriate, the special identifier tag names appear.

For Header records for the **ListTokenInfoByField**, the following field identifiers are included if their associated data is requested:

User information:

- chLastName, chFirstName, chDefaultLogin, chDefaultShell, bTempUser, dateStart, dateEnd

User extension data:

- *tagBUED*,ExtnKey,ExtenData

Group Membership Group names:

- *tagBGM*,GrpName

Group Membership Group values:

- *tagBGM*,GrpDefLogin,GrpDefShell,GrpName,SecurityDomainName

  The options for Group membership Group names and Group membership Group values are mutually exclusive.

Token information:

- *tagBTOK*,chSerialNum, iInterval, dateBirth, dateDeath, dateLastLogin, iType, bHex, bEnabled, bNewPINMode, bMustCreatePIN, iNextCodeStatus, iBadTokenCodes, datePIN, dateEnabled, dateCountsLastModified, bProtected, bDeployed, iCount,notes

Token extension data:

- *tagBTED*,ExtnKey,ExtnData

  If Token information and Token extension data is requested, any data appears in pairs for as many assigned Tokens a User has. For example, a User with two assigned Tokens, each with extension data, has a listing appended to any of the above requested information similar the following:

  tagBTOK, *token info*,*tagBTED*,extnKey1,extnData1, *tagBTOK*, *token info*,*tagBTED*,extnKey1,extnData1,extnKey2,extnData2

MiscVariable is used to correct extension keys and/or data that contain commas. Because the retrieval of the extension data and keys is returned as a comma separated list, it can be difficult to differentiate between the key and the data. When requesting extension data in this list, the AMBA utility attempts to correct key/data pairs that contain more than one comma. It parses the results and builds a key with everything up to the first comma. If the key is valid, it is assumed that the remaining commas are in the data. If the key is not valid, the next section of the data up to the next comma is appended to the previous key and the process is repeated.

When a key is found, all extra commas in the key and data are either deleted or replaced as described below. There is no guarantee that the key is the correct one, because it is possible to build keys containing commas that is valid as either the first portion only, or any concatenated portions. In any case, **MiscVariable** can be used to control the substitution.

MiscVariable can contain the following values:

Delete	delete all extra commas in the key/data pairs
Space	replace all extra commas with a space character
char	replace all extra commas with the provided character

The default action is to replace extra commas with semi-colons.

If **IdentitySource** and **SecurityDomain** are not provided, the search for the User is carried out in the default realm. The results file displays the security domain for each user.

List User Information for User

Action	**LUI**
Required Fields	DefLogin or TokSerial
Optional Fields	OutputOption, ExtnDataOption, MiscVariable, SecurityDomain, IdentitySource

Produces User and assigned Token information for one User. This action duplicates the List User Information by Field action in every respect except this action supplies the information for a single supplied default login or an associated User if a Token serial number is supplied. If both are supplied, the **TokSerial** is ignored.

If **DefLogin** is provided, and **SecurityDomain** and **IdentitySource** are not provided, the search for the User is carried out in the default realm.

If **TokSerial** is provided, **SecurityDomain** and **IdentitySource** are not required and are ignored if provided.

Consult the **LUIF** action for the definition of the optional fields usage and output formatting options.

List Token Information by Field

Action	**LTIF**
Required Fields	None
Optional Fields	CompareField, CompareType, CompareValue, OutputOption, ExtnDataOption, MiscVariable, SecurityDomain, IdentitySource

Produces a list of Token serial numbers or a list of Token information for each Token serial number. The requested information is written to the Results File.

The Token serial numbers selected are based on three input parameters, **CompareField**, **CompareType**, and **CompareValue**. The listing is controlled by three additional parameters, **OutputOption**, **ExtnDataOption**, and **MiscVariable**.

If a selected Token is assigned to a User, User information and Group information can be included in the output report. For example, this action can be used to report a list of Users who have Tokens expiring in 45 days along with the User and User extension data for this Token. The resulting report can be used to generate a mailing list or to automatically notify Users via email. This depends on the availability of information in User extension data fields.

If **SecurityDomain** is not provided, the search for the Token is carried out in the default security domain. Token searches automatically include all sub domains. The results file displays the security domain for each token.

If an **IdentitySource** is not provided, listings containing User information, **OutputOption** 10 and higher, are listed for each **IdentitySource**.

If an **IdentitySource** is provided then listings containing User information are limited to Users existing in that **IdentitySource**. To limit User information to the default **IdentitySource**, the **IdentitySource** must be explicitly declared.

IdentitySource must be provided if the specified **SecurityDomain** is mapped to a different realm other than the default. In this case, listings for reports containing User information, **OutputOption** 10 and higher, the listings are limited to the specified **IdentitySource**.

Select the **CompareField** and the **CompareType** values from the following and supply a value for **CompareValue** of the type listed in the value column.

Token Listed	Field	Type	Value
All	0	0	Ignored
By assignment	1		
All assigned Tokens		1	Ignored
All unassigned Tokens		2	Ignored

Token Listed	Field	Type	Value
By Token type	2		
All assigned Tokens of specific type		1	Token type*
All unassigned Tokens of specific type		2	Token type*
All Tokens of specific type		3	Token type*
All assigned Tokens for specific software token device type		4	Software token device type**
All unassigned Tokens for specific software token device type		5	Software token device type**
All Tokens for specific software token device type		6	Software token device type**
By replacement	3		
All assigned original Tokens		1	Ignored
All assigned replacement Tokens		2	Ignored
By expiration	4		
All expired Tokens		1	Ignored
All assigned expired Tokens		2	Ignored
All expired Tokens and Tokens expiring within a number of days		3	Number of Days
All expired assigned Tokens and assigned Tokens expiring within a given number of days		4	Number of days
All Tokens set to expire		5	Number of Days
All assigned Tokens set to expire		6	Number of Days
By status	5		
All assigned and disabled		1	Ignored
All assigned and enabled		2	Ignored
All with cleared Pins		3	Ignored
All in New PIN Mode		4	Ignored
All in Next Token Code mode		5	Ignored
All lost Tokens		6	Ignored
All with lost status expired		7	Ignored

Token Listed	Field	Type	Value
By status (continued)	5		
All with lost status expired, or set to expire in a number of days		8	Number of Days
All with lost status set to expire in a number of days		9	Number of Days
By Token extension	6		
All that have extension records		1	Ignored
All that do not have extension records		2	Ignored
All that have extension records with the provided key		3	String
All that have extension records without a provided key		4	String

* For Token type values see TokenType in Input Field Definitions:
** Software token device type is specified in Software Token Device Type Specifications:

Table 19: Software Token Device Type Specifications

Specification	Description
Empty	All software token device types qualify.
“None”	(no quotes) Software tokens with no device type specification qualify.
FamilyKey	Software token device types with matching name and optional version qualify.
Name **[:** [version]]	Name only will filter all software token device types that match name regardless of version.
Name**:**	Same as name only.
Name**:***version*	Filters software token device types that match name and version.
Examples:	
BlackBerry BlackBerry**:**3.0 BlackBerry **:** 3.0 Web SDK **:**	any version version 3.0 only same as proceeding example any version.

Case is significant for all name matches.

OutputOption can contain the following values:

[[-31 | -21 | -11 | -1 | 0 | 10 - 17 | 20 - 27] [H | h] [A | a]]

For the definitions of the **OutputOption** values, refer to OutputOption Values.

Default for all versions is 0. List Token information for all selected Tokens, do not write a header record. In the definitions below, User information and extended User information is listed only if the selected Token is assigned.

If an Append mode is not specified, then new is assumed. Append and Header can be used simultaneously, which can result in header lines within the file body. This is intentional and it is up to the user to determine which options make sense and whether or not down stream applications are able to correctly process this file.

Because **CIF** actions can be used between List actions, data columns might not line up if the append option is used. This action is intentional and the user must choose the options appropriate for the desired outcome.

The **ExtnDataOption** can contain the following values:

0 - no additional data listed	6 - option 2 and 4
1 - include Token extension data	7 - option 3 and 4
2 - include User extension data	8 - include Group membership field values
3 - option 1 and 2	9 - option 1 and 8
4 - include Group membership Group names	10 - option 2 and 8
5 - option 1 and 4	11 - option 3 and 8

Because these options can appear zero or more times, formatting tags are inserted to assist in parsing the data. If Token or User extension data appears in the output, the first extension data key is preceded by *tagBTED* for Token extension data and *tagBUED* for User extension data. The tag is followed by as many extension key/data pairs as necessary. If either or both (Token or User) have no extension key/data pairs, a tag will not appear. If Group membership data appears in the output, the first Group membership name or the first Group membership field is preceded by the tag *tagBGM*. If there are no Group membership values, the tag will not appear.

If a header record has been requested, appropriate field names are written for each option. The field names will appear only once, even though the actual data can appear more than once. When appropriate, the special identifier tag names will also appear.

For Header records for the **ListTokenInfoByField,** the following field identifiers are included if their associated data has been requested:

Token information:

chSerialNum, iInterval, dateBirth, dateDeath, dateLastLogin, iType, bHex, bEnabled, bNewPINMod, iNextCodeStatus, iBadTokenCodes, datePIN, dateEnabled, dateCountsLastModified, bProtected, bDeployed, iCount,notes

CT-KIP information:

CtkipActivationCode, CtkipTriggerURL, PrincpalID, SecurityDomainID, CtkipKeyLastDownloadedBy, CtkipKeyLastDownloadedOn, IsAssighend, IsEnabled, IsGroupActivationCode, Notes

User information:

chLastName, chFirstName, chDefaultLogin, bMustCreatePIN, chDefaultShell, bTempUser, dateStart, dateEnd

Token extension data:

tagBTED,ExtnKey,ExtnData

User extension data:

tagBUED,ExtnKey,ExtenData

Group Membership Group names:

tagBGM,GrpName

Group Membership Group values:

tagBGM,GrpDefLogin,GrpDefShell,GrpName,SecurityDomainName

The options for Group membership, Group names, and Group membership Group values are mutually exclusive.

MiscVariable is used to correct extension keys and/or data that contain commas. Because the retrieval of the extension data and keys is returned as a comma separated list, it can be difficult to differentiate between the key and the data. When requesting extension data in this list, the AMBA utility attempts to correct key/data pairs that contain more than one comma. It parses the results and builds a key with everything up to the first comma. If the key is valid, it is assumed that the remaining commas are in the data. If the key is not valid, the next section of the data up to the next comma is appended to the previous key and the process is repeated. When a key is found, all extra commas in the key and data are either deleted or replaced as described below. There is no guarantee that the key is the correct one, because it is possible to build keys containing commas that is valid as either the first portion only, or any concatenated portions. In any case, **MiscVariable** can be used to control the substitution.

MiscVariable can contain the following values:

Delete	delete all extra commas in the key/data pairs
Space	replace all extra commas with a space character
char	replace all extra commas with the provided character

The default action is to replace extra commas with semi-colons.

List Token Information

Action	**LTI**
Required Fields	TokSerial
Optional Fields	OutputOption, ExtnDataOption, MiscVariable, SecurityDomain, IdentitySource

Produce Token information and User information if the Token is assigned. It duplicates List Token Information by Field in every respect except that it supplies the information for a single supplied Token, identified by TokSerial.

Consult the **LTIF** action for the definition of the optional fields, usage and output formatting options.

List Token Summary Report

Action	**LTSR**
Required Fields	
Optional Fields	MiscVariable, SecurityDomain, OutputOption

Produces a Token summary report with two listings. The first listing consists of an enumeration of each Token type in the database and a count of the number of Tokens of that type. The second listing is an enumeration of the Token device types. Both listings list an item only if one or more items are declared in the database.

MiscVariable is used to select Token state. supported values for this action are:

1 Report on assigned Tokens only

2 Report on unassigned Tokens only

3 Report on all Tokens.

The **SecurityDomain** is used to specify at what level to begin the search.

The supported values for **OutputOption** in this action are:

empty	Create a new file if one does not exist.
A | a	Append output to an existing results file.

List Secondary Nodes for Agent Host

Action	**LSN**
Required Fields	AgentHostName
Optional Fields	SecurityDomain, IdentitySource

Lists all of the secondary nodes associated with a given Agent Host. The list is written to the Results File.

If **SecurityDomain** is not provided, then the search for the Agent Host is carried out in the default security domain.

IdentitySource must be provided if the specified **SecurityDomain** is mapped to a different realm other than the default.

9 Multiple Token Actions

The multiple Token actions are provided as a means of assigning a Token or issuing a replacement Token to a large number of Users. There are some limitations to these actions.

If software tokens are assigned, the **-g** and **-gdir** Command-line Options can be used to instruct the AMBA utility to generate Software Token Database Files and place them in the specified directory. If the **-g** or **--ctkip** options are not used, these actions assign Tokens but do not build any output files or generate any CT-KIP credentials. The Set Software Token Profile action can be used to force a specific device type for credential generation.

If a database is properly populated with extension data, the output of these actions can be used for Automatic Notification. The output of these actions is a comma-separated variable list that can be processed by most scripting languages, spreadsheet programs, or word processor programs.

If the report is redirected to **STDOUT** or **STDERR**, the last line of output is the single word 'Done' followed by a newline sequence. A report that generates no output consists of a single line beginning with the word 'Done'.

To improve efficiency these actions process Token assignments in groups of 500. A side effect of this is that only one log entry is made at the completion of the action. Therefore there is no individual listing for each Token assignment. If this information is desired, then the appropriate **OutputOption** should be configured. The output information is produced before each group of 500 operations so if an error occurs during the processing of a group of 500, the output list would most likely be incorrect. It is left to the user to determine where in the process the error occurred and take appropriate action. For the definitions of the **OutputOption** values, refer to OutputOption Values.

This chapter provides a description of the Multiple Token actions, and details the required and optional fields. Where fields are not described, see the description provided in the Input Field Definitions.

The topics in this chapter include:

- Multiple Softtoken Deployment
- Multiple Token Assignment
- Multiple Token Disable/Rescind
- Multiple Token Replacement

Multiple Softtoken Deployment

Action	**MSD**
Required Fields	RangeMode
Optional Fields	StartRange, EndRange, Password, Filename, CopyProtect, OverOption, IdentitySource, SecurityDomain, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, NicknameIsCtkipCode, DeviceserialIsCtkipCode

Deploys Tokens in Software Token Database Files which are stored in zip files.

Note: The Tokens in a given range must be of the same device type.

This action specifies a range of Tokens and deploys them to assigned Users specified by serial number or default login. Prior to calling this function, all Users must exist in the RSA Authentication Manager (Authentication Manager) database, and all Tokens included in the range must be assigned to a User.

RangeMode specifies criteria used to deploy assigned software tokens either by serial number or User default login.

The following table lists acceptable values for use with **rangeMode**, the affect each value has on subsequent fields, and the process completed when the function is called.

RangeMode	StartRange	EndRange	Process
0	**TokSerial**	Ignored	Deploys one assigned Token specified by the serial number.
1	Ignored	Ignored	Deploys all assigned Tokens.
2	**TokSerial**	**TokSerial**	Deploys all assigned Tokens within the specified range of serial numbers.
3	**DefLogin**	Ignored	Deploys all Tokens that are assigned to the specified User.
4	**DefLogin**	**DefLogin**	Deploys all Tokens that are assigned to the range of Users specified.

StartRange specifies the first Token serial number or User default login in a range.

EndRange specifies the last Token serial number or User default login in a range.

Password specifies the administrator password to access an SDTID file. If there is no password associated with the file, an empty string can be passed.

FileName specifies the name of the software token database zip file.

IdentitySource is used to validate the security domain provided:

- If **RangeMode** is set to 0, 1, or 2, this field validates the security domain provided. For these values, Tokens are fetched based on the **SecurityDomain**.
- For **RangeMode** 3 and 4, the search for the Users is carried out in the given identity source. If not provided, the search for the Users is carried out in the default realm.

IdentitySource must be provided if the **SecurityDomain** provided is mapped to a different realm other than the default.

SecurityDomain specifies the location of the search for the Tokens. If not provided, the search for the Tokens is carried out in the default security domain.

Nickname, **DeviceSerialNumber**, **SiteFile**, **SiteURL1**, **SiteURL2**, and **SiteURL3** assign values to Software Token Notifications and Attributes.

MSD Example

The following is an example of the **MSD** action:

```
Action,RangeMode,StartRange,EndRange,password,FileName,CopyProtect,
OverOption,IdentitySource,SecurityDomain,Nickname
msd,2,000027050103,000027050109,password,SerialRange.zip,1,0,,
msd,4,user1,user50,, UserRange.zip,1,1,,
msd,0, 000027050110,password,SerialRange.zip,1,0,-1
```

The first input line creates the file **SerialRange.zip** containing the range of Tokens from 000027050103 to 000027050109 in the form of a software Token database file.

The second input line creates the file **UserRange.zip** containing the range of Tokens associated with the Users from User 1 to User 50 in the form of an SDTID file.

The last line appends Token 000027050110, in the form of an SDTID file, to the **SerialRange.zip** file.

Multiple Token Assignment

Action	**MTA**
Required Fields	None
Optional Fields	CompareField, OutputOption, ExtnDataOption, MiscVariable**,** TokEnabled, SetPin, PinMode, PinType, SoftIDParams, SoftIDPW, IdentitySource, SecurityDomain, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, NicknameIsCtkipCode, DeviceserialIsCtkipCode

This action is used to scan the database for Users that do not have a Token or password currently assigned and who are not flagged as a temporary, disabled or remote User. If a User matches these conditions, a Token is assigned. The type of RSA SecurID (SecurID) Token to be assigned is determined by the value of **CompareField**. When the Token is assigned, its PIN is cleared and the Token is disabled. A listing is generated. The list content is determined by the value of **OutputOption**. The listing indicates the Token serial number assigned and the User to which it's assigned to. The listing can be useful for automated notification or mailing label generation.

MTA can be used to assign values to Software Token Device Type Attributes.

CompareField is used to indicate what type of Token to assign. Accepted values are:

-1	Any unassigned Token (default)	5	SecurID Smartcard
0	SecurID Standard Card	6	SecurID Modem
1	SecurID PINPAD Card	7	SecurID Crypto
2	SecurID Key Fob	8	SecurID Proteus
3	SecurID Watch	9	SecurID USBCOSMO (SID800)
4	SecurID software token	10	SecurID Flextoken

There must be enough Tokens of the desired type imported into the database to satisfy the number of Users needing a Token assigned. If there are not enough unassigned Tokens in the database, the process runs until the last available Token is assigned. The process terminates and an appropriate entry is recorded in the log.

OutputOption is used to determine what information is included in the output report for each Token assigned. If extension data fields have been appropriately initialized, they can be utilized in an automated notification system or for automated mailing label generation. The report contains one line for each assigned Token and is in CSV format. Most scripting languages as well as document and spreadsheet editors can parse these files and generate the desired end format.

OutputOption can contain:

[[0 | [1 - 4, 10 - 14]] [h | H] [r | R] [a | A]]

where:

0	TokSerial, DefLogin, FirstName, LastName (default)
10	TokSerial + *UserInfo*
UserInfo	chLastName, chFirstName, chDefaultLogin, bCreatePIN, bMustCreatePIN, chDefaultShell, bTempUser, dateStart, dateEnd

For the definitions of the standard **OutputOption** values, refer to OutputOption Values.

If the Append mode is not specified, then new is assumed. Append and Header can be used simultaneously, which can result in heading lines within the file body. This is intentional and it is up to the user to determine which options make sense and whether or not down stream applications are able to correctly process this file. Also, because **CIF** actions can be used between multiple actions, if the Append option is used, data columns might not line up. This is intentional and the user must choose the options appropriate for the desired outcome.

ExtnDataOption is used to indicate whether or not extension data should be included in the output report, and can contain the following values:.

0 - no additional data listed	6 - option 2 and 4
2 - include User extension data	8 - include Group membership field values
4 - include Group membership Group names	

Because these options can appear zero or more times, formatting tags are inserted to assist in parsing the data. If Token or User extension data appears in the output, the first extension data key is preceded by *tagBTED* for Token extension data and *tagBUED* for User extension data. The tag is followed by as many extension key/data pairs as necessary. If either or both Token or User has no extension key/data pairs, a tag does not appear. If Group membership data appears in the output, the first Group membership name or the first Group membership field is preceded by the tag *tagBGM*. If there are no Group membership values, the tag does not appear.

The following field identifiers are included in the heading, if their associated data has been requested:

Token extension data:

tagBTED,ExtnKey,ExtnData

User extension data:

tagBUED,ExtnKey,ExtenData

Group Membership group names:

tagBGM,GrpName

Group Membership group values:

tagBGM,GrpDefLogin,GrpDefShell,GrpName,SiteName

If a header record is requested, appropriate field names are written for each option. The field names appear only once, even though the actual data can appear more than once. When appropriate, the special identifier tag names also appear.

MiscVariable is used to control extension data comma replacement. Many users have included commas in extension data. The Authentication Manager custom API, used by the AMBA utility to interface to the database, can sometimes incorrectly parse this data because it returns the data in a comma delimited list. In certain instances, the API cannot distinguish between API list separators and commas contained in the data. This field specifies the replacement or deletion of the extension data commas. The replacement, if requested, only affects the generated report and is never saved in the database. Supported values for this field are:

char	character to replace embedded extension key/data commas
space	replace commas with a space
delete	delete commas

If no value is supplied in **MiscVariable**, the default process is to replace embedded commas with the semi-colon (;) character.

SetPin is used to initialize the PIN of the newly assigned Token:

0	Clear PIN (default). Automatically puts the Token in New PIN Mode.
literal	Any other value sets all newly assigned Tokens to this PIN.

IdentitySource specifies the location of the search for the range of Users. If not provided, the search for the Users is carried out in the identity source mapped to the default security domain.

SecurityDomain specifies the location of the search for the range of Tokens. If not provided, the search for the Tokens carried out in the default security domain.

Multiple Token Disable/Rescind

Action	**MTD**
Required Fields	CompareType
Optional Fields	CompareField, CompareValue, OutputOption, ExtnDataOption, MiscVariable, IdentitySource, SecurityDomain

This action is used to scan the database for assigned Tokens that have not been used for a specified period of time and disable or rescind them. The database is scanned for assigned Tokens and the last logon date is compared with the cutoff date. The cutoff value is supplied in the **CompareValue** field. The Token types scanned can be all, or a specific type determined by the **CompareField** value. The listing indicates the Token serial number being disabled/rescinded and the User to which it's assigned. Additional listing information is available as noted below. The listing can be useful for automated notification or mailing label generation.

CompareType designates the type of process to perform.

1 - Disable inactive Tokens (include last logon date of 1/1/1986)

2 - Disable inactive Tokens (exclude last logon date of 1/1/1986)

3 - Rescind inactive Tokens (include last logon date of 1/1/1986)

4 - Rescind inactive Tokens (exclude last logon date of 1/1/1986)

Note: When a Token is imported into the system, the last logon date is initialized to 01/01/1986, the birth date of RSA. To isolate a Token that has not been used yet from one that has been used but not for some period of time, use option 2 or 4.

CompareField is used to indicate what type of Token to scan for. Accepted values are:

-1	Any unassigned Token (default)	5	SecurID Smartcard
0	SecurID Standard Card	6	SecurID Modem
1	SecurID PINPAD Card	7	SecurID Crypto
2	SecurID Key Fob	8	SecurID Proteus
3	SecurID Watch	9	SecurID USBCOSMO (SID800)
4	SecurID software token	10	SecurID Flextoken

CompareValue is used to set the cutoff date to determine inactivity. Valid entries are:

0 or empty	Cutoff date is (current date - 90) days
1 -365	Cutoff date is set to (current date - number).
MM/dd/yyyy	A cutoff date in the given format. The date must be greater than or equal to the (current date - 365 days) and less than the current date.

OutputOption is used to determine the information that is included in the output report for each disabled Token. If extension data fields have been appropriately initialized, they can be utilized in an automated notification system or for automated mailing label generation. The report contains one line for each disabled/rescinded Token and is in CSV format. Most scripting languages as well as document and spreadsheet editors can parse these files and generate the desired end format.

Supported values for this field are:

[[0 | [1 - 4, 10 - 14]] [h | H] [r | R] [a | A]]

where:

0	TokSerial, DefLogin, FirstName, LastName (default)
10	TokSerial + *UserInfo*
UserInfo	chLastName, chFirstName, chDefaultLogin, bCreatePIN, bMustCreatePIN, chDefaultShell, bTempUser, dateStart, dateEnd

For the definitions of the standard **OutputOption** values, refer to OutputOption Values.

If the Append mode is not specified, then new is assumed. Append and Header can be used simultaneously, which can result in heading lines within the file body. This is intentional and it is up to the user to determine which options make sense and whether or not down stream applications are able to correctly process this file. Also, because **CIF** actions can be used between multiple actions, if the Append option is used, data columns might not line up. This is intentional and the user must choose the options appropriate for the desired outcome.

ExtnDataOption can contain the following values:

0 - no additional data listed	6 - option 2 and 4
1 - include Token extension data	7 - option 3 and 4
2 - include User extension data	8 - include Group membership field values
3 - option 1 and 2	9 - option 1 and 8
4 - include Group membership Group names	10 - option 2 and 8
5 - option 1 and 4	11 - option 3 and 8

Because these options can appear zero or more times, formatting tags are inserted to assist in parsing the data. If Token or User extension data appears in the output, the first extension data key is preceded by *tagBTED* for Token extension data and *tagBUED* for User extension data. The tag is followed by as many extension key/data pairs as necessary. If either or both Token or User has no extension key/data pairs, a tag does not appear. If Group membership data appears in the output, the first Group membership name or the first Group membership field is preceded by the tag *tagBGM*. If there are no Group membership values, the tag does not appear.

If a header record has been requested, appropriate field names are written for each option. The field names appear only once, even though the actual data can appear more than once. When appropriate, the special identifier tag names also appear.

MiscVariable is used to control extension data comma replacement. Many users have included commas in extension data. The Authentication Manager custom API, used by the AMBA utility to interface to the database, can sometimes incorrectly parse this data because it returns the data in a comma delimited list. The API might not distinguish between API list separators and commas contained in the data. This field specifies the replacement or deletion of the extension data commas. The replacement, if requested, only affects the generated report and is never saved in the database. Supported values for this field are:

char	character to replace embedded extension key/data commas
space	replace commas with a space
delete	delete commas.

If no value is supplied in **MiscVariable**, the default process is to replace embedded commas with the semi-colon (;) character.

IdentitySource is used only to validate the security domain provided. If not provided, the default security domain, for example, the **Internal DataBase** is used. **IdentitySource** must be provided if the specified **SecurityDomain** is not mapped to the default. Processing this action is purely driven by the Security Domain value.

SecurityDomain specifies the location of the search for the range of Tokens. If not provided, the Tokens are searched in the default security domain, for example, the **System Domain** created during installation.

Multiple Token Replacement

Action	**MTR**
Required Fields	None
Optional Fields	CompareField, CompareType, CompareValue, OutputOption, ExtnDataOption, MiscVariable, MinTokenLife, TokenSerial, TokEnabled, PinMode, PinType, SoftIDParams, SoftIDPW, IdentitySource, SecurityDomain, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, NicknameIsCtkipCode, DeviceserialIsCtkipCode

This action is used to scan the database for Tokens that have expired, are about to expire or both, determined by **CompareType** and **CompareValue**. An input parameter is used to indicate whether or not the associated User can have additional Tokens. If the User qualifies and is not flagged as a temporary or remote User, a replacement Token is assigned. The type of Token to be assigned is determined by the value of **CompareField**. When the Token is assigned, its PIN is cleared and the Token is disabled. A listing is generated. The list content is determined by the value of **OutputOption**. The listing indicates the Token serial number assigned and the User to which it's assigned. The listing can be useful for automated notification or mailing label generation.

You can use **MTR** to assign values to software token device type attributes. For more information, see Software Token Device Type Attributes.

The default output for **MTR** is a zip file titled **SoftwareTokens.zip**.

- If the **OutputOption** extract files option (E | e) is included in the relevant action:
  - the **SoftwareTokens.zip** file is unzipped into the same directory and the **SoftwareTokens.zip** file is deleted.
  - one of the file naming options can be included. The U | u option changes the file name to ***User ID*.sdtid** and the T | t option changes the file name to ***token serial*.sdtid**.
- If the extract files option is not included, the file naming options are ignored.

CompareField is used to indicate what type of Token to assign. Accepted values are:

-2	Same as Token being replaced	5	SecurID Smartcard
-1	Any unassigned Token	6	SecurID Modem
0	SecurID Standard Card	7	SecurID Crypto
1	SecurID PINPAD Card	8	SecurID Proteus
2	SecurID Key Fob	9	SecurID USBCOSMO (SID800)
3	SecurID Watch	10	SecurID Flextoken
4	SecurID software token		

For **CompareField** values equal to -1 or -2, any expired or about to expire assigned Token is assigned a replacement Token. For all other values, only assigned Tokens with a matching Token type are assigned a replacement Token.

There must be enough Tokens of the desired type imported into the database to satisfy the relevant number of Users. If there are insufficient unassigned Tokens in the database, the process runs until the last available Token is assigned, the process terminates and an appropriate entry is recorded in the log.

CompareType determines the range to be used for searching for expired Tokens:

0 Replace Tokens already expired (default)

1 Replace Tokens expiring in a number of days

2 Options 0 and 1.

CompareValue provides a base time in which a Token expires:

0 (default)

days from current date

MM/dd/yyyy

MinTokenLife is used to guarantee that a token has a minimum number of days before it expires. It is ignored if **MiscVariable** is set to -1.

OutputOption is used to determine the information that is included in the output report for each Token assigned. If extension data fields have been appropriately initialized, they can be utilized in an automated notification system or for automated mailing label generation. The report contains one line for each assigned Token and is in CSV format. Most scripting languages as well as document and spreadsheet editors can parse these files and generate the desired end format. **OutputOption** can contain the following values:

[[0 | [1 - 4] | [10 - 14]] [h | H] [r | R] [a | A] [u | U] [t | T] [e | E]]

where:

0 Base Information only (default)

10 Base Information + *UserInfo*

For the definitions of the standard **OutputOption** values, refer to OutputOption Values.

Base Information:

SerialNum_O, SerialNum_R, Type, NewPin, DefaultLogin, FirstName, LastName

These are the header names for the expired Token serial number, the replacement Token serial number, the replacement Token type, the User default login, the User first name and the User last name.

If the Append mode is not specified, then new is assumed. Append and Header can be used simultaneously, which can result in heading lines within the file body. This is intentional and it is up to the user to determine which options make sense and whether or not down stream applications are able to correctly process this file. Also, because **CIF** actions can be used between **MTR** actions, if the Append option is used, data columns might not line up. This is intentional and the user must choose the options appropriate for the desired outcome.

ExtnDataOption is used to indicate whether or not extension data is included in the output report. Supported values for this action are:

0 - no additional data listed	6 - option 2 and 4
2 - include User extension data	8 - include Group membership field values
4 - include Group membership Group names	10 - option 2 and 8

MiscVariable is used to control extension data comma replacement. Many users have included commas in extension data. The Authentication Manager custom API, used by the AMBA interface to interface to the database, can sometimes incorrectly parse this data because it returns the data in a comma delimited list. The API might not distinguish between API list separators and commas contained in the data. This field specifies replacement or deletion of the extension data commas. The replacement, if requested, only affects the generated report and the result is never saved in the database.

char	character to replace embedded extension key/data commas
space	replace commas with a space
delete	delete commas.

If no value is supplied in **MiscVariable**, the default action is to replace embedded commas with the semi-colon (;) character.

IdentitySource is used only to validate the security domain provided. If not provided, then the default security domain, for example, the **Internal DataBase** is used. It must be provided if **SecurityDomain** is specified and is not mapped to the default. Processing of this action is purely driven by the Security Domain value.

SecurityDomain specifies the location for the search for the range of Tokens. If not provided, then search is carried out in the default security domain, for example, the **System Domain**.

10 Troubleshooting

The topics in this chapter include:

java.lang.ClassNotFoundException:

The following error message is almost always caused by either an improper AMBA utility installation or an incorrect license file:

```
API return: java.lang.ClassNotFoundException:
Failed to load class
com.rsa.ucm.principal.tools.SearchPrincipalsABACommand
Failed to load class
com.rsa.ucm.principal.tools.SearchTokensABACommand
```

The AMBA utility requires a valid Enterprise Edition or Premium Edition license file, unless you have upgraded to RSA Authentication Manager (Authentication Manager) 8.4 with a Base Server license that includes the AMBA utility add-on option.

Use the **--lic** option to provide the license file to AMBA. For instructions, see Execute the AMBA Utility Command-line Options.

In some instances, it might be necessary to clear the cache. For instructions, see Flush the Cache.

Required Patch Level

RSA recommends that you apply the most recent patches for Authentication Manager.

Flush the Cache

Flush the cache to remove old information from memory. After the cache is flushed, each object is refreshed from the database the next time it is accessed.

Procedure

1. Log on to the Operations Console with one of the following URLs:
   https://*fully qualified domain name*/oc
   https://*fully qualified domain name*:7072/operations-console
2. Click **Maintenance** > **Flush Cache**.
3. If prompted, enter your Super Admin User ID and password, and click **OK**.
4. Under Flush Cache, select **Flush all cache objects** to flush all the caches.
5. Click **Flush**.

11 Samples

The following samples provide example data files for the AMBA utility.

To use these example files, copy them into a text editor and save each file with the .csv extension. Open the files using a spreadsheet program and modify the data.

For more information about preparing a data file, see Prepare the Data File.

This chapter includes the following topics:

Create New Users, Assign Software Tokens, Specify Software Token Profiles, and Provision Tokens

This example creates 10 new Users, assigns tokens, specifies software token profiles for each User, Android 2.x ctkip and iOS20 ctkip url, and provisions the tokens through CT-KIP over email, SMTP.

```
Action,LastName,DefLogin,TokEnabled,UserPwd,MiscVariable,SoftTokenProfile,Email,
DeliveryMethod
AUTA,autauser1,autauser1,1,password$,4,Andoid 2.x ctkip,autauser1@test.com,SMTP
AUTA,autauser2,autauser2,1,password$,4,iOS20 ctkip url,autauser2@test.com,SMTP
AUTA,autauser3,autauser3,1,password$,4,Android 2.x ctkip,autauser3@test.com,SMTP
AUTA,autauser4,autauser4,1,password$,4,Android 2.x ctkip,autauser4@test.com,SMTP
AUTA,autauser5,autauser5,1,password$,4,iOS20 ctkip url,autauser5@test.com,SMTP
AUTA,autauser6,autauser6,1,password$,4,Android 2.x ctkip,autauser6@test.com,SMTP
AUTA,autauser7,autauser7,1,password$,4,Android 2.x ctkip,autauser7@test.com,SMTP
AUTA,autauser8,autauser8,1,password$,4,iOS20 ctkip url,autauser8@test.com,SMTP
AUTA,autauser9,autauser9,1,password$,4,iOS20 ctkip url,autauser9@test.com,SMTP
AUTA,autauser10,autauser10,1,password$,4,iOS20 ctkip url,autauser10@test.com,SMTP
```

For details about the **AUTA** action, see Add User and Token Automatic.

Replace and Provision Tokens

This example replaces software tokens for 10 users, enables the tokens, sets the tokens to New PIN Mode, provisions the tokens through a password-protected Software Token Database File, and sets a minimum token life of 120 days.

```
Action,TokSerial,MiscVariable,TokEnabled,SetPin,IdentitySource,SecurityDomain,
DeviceSerialNumber,DeliveryMethod,SoftIDPW,SoftIDParams,MinTokenLife
REPTA,132251663,4,1,C,InternalDatabase,SystemDomain,S-111000555,SMTP2,password,001,120
REPTA,132251664,4,1,C,InternalDatabase,SystemDomain,S-111000555,SMTP2,password,001,120
REPTA,132251665,4,1,C,InternalDatabase,SystemDomain,S-111000555,SMTP2,password,001,120
REPTA,132251666,4,1,C,InternalDatabase,SystemDomain,S-111000555,SMTP2,password,001,120
REPTA,132251667,4,1,C,InternalDatabase,SystemDomain,S-111000555,SMTP2,password,001,120
REPTA,132251668,4,1,C,InternalDatabase,SystemDomain,S-111000555,SMTP2,password,001,120
REPTA,132251669,4,1,C,InternalDatabase,SystemDomain,S-111000555,SMTP2,password,001,120
REPTA,132251670,4,1,C,InternalDatabase,SystemDomain,S-111000555,SMTP2,password,001,120
REPTA,132251671,4,1,C,InternalDatabase,SystemDomain,S-111000555,SMTP2,password,001,120
REPTA,132251672,4,1,C,InternalDatabase,SystemDomain,S-111000555,SMTP2,password,001,120
```

For details about the **REPTA** action, see Replace Token Automatic.

Change User and Token Domains

This example changes 10 users and their assigned tokens from the “Sales” Security Domain to the “Engineering” Security Domain.

```
cif,Action,DefLogin,DestinationSecurityDomain,MiscVariable,IdentitySource,SecurityDomain
CUSD,cusduser,Engineering,0,Internal Database,Sales
CUSD,cusduser1,Engineering,0,Internal Database,Sales
CUSD,cusduser2,Engineering,0,Internal Database,Sales
CUSD,cusduser3,Engineering,0,Internal Database,Sales
CUSD,cusduser4,Engineering,0,Internal Database,Sales
CUSD,cusduser5,Engineering,0,Internal Database,Sales
CUSD,cusduser6,Engineering,0,Internal Database,Sales
CUSD,cusduser7,Engineering,0,Internal Database,Sales
CUSD,cusduser8,Engineering,0,Internal Database,Sales
CUSD,cusduser9,Engineering,0,Internal Database,Sales
```

For details about the **CUSD** action, see Change User/Token Security Domain.

Add Agent Hosts

This example adds 10 Agent Hosts that are unrestricted and enabled. 5 agent hosts are Standard Agents, and 5 are Web Agents.

```
Action,AgentHostName,AgentHostAddress,AgentHostType,AgentRestriction,EnableFlag,
SecurityDomain,Operation,AgentTrustRealm
AAH,win-rsa.vcloud.local,198.75.63.22,1,0,TRUE,BusinessDomain,ADD,0
AAH,win22-rsa.vcloud.local,198.75.63.25,1,0,TRUE,BusinessDomain,ADD,0
AAH,win23-rsa.vcloud.local,198.75.63.26,1,0,TRUE,BusinessDomain,ADD,0
AAH,win24-rsa.vcloud.local,198.75.63.27,1,0,TRUE,BusinessDomain,ADD,0
AAH,win25-rsa.vcloud.local,198.75.63.28,1,0,TRUE,BusinessDomain,ADD,0
AAH,win26-rsa.vcloud.local,198.75.63.122,2,0,TRUE,BusinessDomain,ADD,0
AAH,win77-rsa.vcloud.local,198.75.63.132,2,0,TRUE,BusinessDomain,ADD,0
AAH,win88-rsa.vcloud.local,198.75.63.102,2,0,TRUE,BusinessDomain,ADD,0
AAH,win99-rsa.vcloud.local,198.75.63.112,2,0,TRUE,BusinessDomain,ADD,0
AAH,win10-rsa.vcloud.local,198.75.63.92,2,0,TRUE,BusinessDomain,ADD,0
```

For details about the **AAH** action, see Add Agent Host.

Enable On-Demand Authentication and Distribute Tokencodes

This example enables On-Demand Authentication for five users, sets a temporary PIN of **1234** for all of the users, and sends the ODA tokencode through email.

```
Action,DefLogin,IdentitySource,PINIndicator,SetPin,DeliveryMethod,OutputOption,SubDomain
EODA,auguser,,SET_TEMP_PIN,1234,SMTP,N,1
EODA,auguser1,,SET_TEMP_PIN,1234,SMTP,N,1
EODA,auguser2,,SET_TEMP_PIN,1234,SMTP,N,1
EODA,auguser3,,SET_TEMP_PIN,1234,SMTP,N,1
EODA,auguser4,,SET_TEMP_PIN,1234,SMTP,N,1
```

For details about the **EODA** action, see Enable OnDemand Authentication.

Change User Login from “firstname” to “firstinitial.lastname”

This example change the User login from “firstname” to “firstinitial.lastname”.

```
Action,DefLogin,MiscVariable,IdentitySource,SecurityDomain
UUD,NewLoginName,OldLoginName,Internal Database,SystemDomain
UUD,f.lastname,firstname,Internal Database,SystemDomain
```

For details about the **UUD** action, see Update User Data.

Software Token Distribution

This example demonstrates the use of a data file with the **AUTA** action to deliver software tokens, provisioned with CT-KIP, by email (SMTP).

This sample data file includes the **SoftTokenProfile** field in each line:

```
Action,LastName,DefLogin,TokEnabled,UserPwd,MiscVariable,SoftTokenProfile,Email,
DeliveryMethod
AUTA,autauser1,autauser1,1,password$,4,Andoid 2.x ctkip,autauser1@test.com,SMTP
AUTA,autauser2,autauser2,1,password$,4,iOS20 ctkip url,autauser2@test.com,SMTP
```

In this sample data file, the **SSTP** action defines a global software token profile followed by the **CIF** action with actions:

```
Action,SoftTokenProfile
SSTP,iOS20 ctkip url
CIF,Action,LastName,DefLogin,TokEnabled,UserPwd,MiscVariable,Email,DeliveryMethod
AUTA,autauser5,autauser5,1,password$,4,autauser5@test.com,SMTP
AUTA,autauser6,autauser6,1,password$,4,autauser6@test.com,SMTP
```

To use the sample data files, run the AMBA utility with the following command, where **auta1.csv** is the data file and **amba.lic** is the license file:

```
./rsautil AMBulkAdmin -a admin -P password --ctkip -i auta1.csv --lic ./amba.lic
```

For details about the **AUTA** action, see Add User and Token Automatic.

For details about the **SSTP** action, see Set Software Token Profile

For details about the **CIF** action, see Change Input Format