RSA® Authentication Manager 8.2 Service Pack 1 Bulk Administration 1.6.0 Custom Application Guide

(AMBA)

Contact Information

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

Trademarks

RSA, the RSA Logo and EMC are either registered trademarks or trademarks of EMC Corporation in the United States and/or other countries. All other trademarks used herein are the property of their respective owners. For a list of RSA trademarks, go to **www.emc.com/legal/emc-corporation-trademarks.htm#rsa**.

License Agreement

This software and the associated documentation are proprietary and confidential to EMC, 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 EMC.

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 EMC software described in this publication requires an applicable software license.

EMC 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." EMC CORPORATION 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.

March 2017

Contents

Preface

About This Guide

This document describes how administrators can use the RSA® Authentication Manager 8.2 Service Pack 1 (SP1) Bulk Administration 1.6.0 (AMBA) utility developed from the RSA Authentication Manager Server Admin APIs. This utility enables administrators to perform administration from the command line or in a background mode through scheduled scripting.

RSA Authentication Manager 8.2 SP1 Documentation

The latest product documentation is always available on RSA Link at **https://community.rsa.com**.

Title	Purpose
Configuration	
Planning Guide	Describes the high-level architecture of Authentication Manager and how it integrates with your network.
Hardware Appliance Getting Started	Describes how to deploy a hardware appliance and perform the Authentication Manager Quick Setup process.
Virtual Appliance Getting Started	Describes how to deploy a virtual appliance and perform the Authentication Manager Quick Setup process.
Setup and Configuration Guide	Describes how to set up and configure Authentication Manager, and how to upgrade from version 8.2 to version 8.2 Service Pack 1.
Security Configuration Guide	Describes the security configuration settings available in RSA Authentication Manager. It also describes secure deployment and usage settings, secure maintenance, and physical security controls.
Administration	
Administrator's Guide	Provides an overview of Authentication Manager and its features. Describes how to configure the system and perform a wide range of administration tasks.

Title	Purpose
Developer's Guide	Provides information about developing custom programs using the RSA Authentication Manager application programming interfaces (APIs). Includes an overview of the Authentication Manager APIs and the related Javadoc. **Note:** The software development kit (SDK) is located in the RSA Authentication Manager SDK directory of the **RSA Authentication Manager 8.2 Service Pack 1 Extras ZIP** file. The **Extras ZIP** file is located on Download Central at **https://download.rsasecurity.com**.
RSA RADIUS Reference Guide	Describes the usage and settings for the initialization files, dictionary files, and configuration files used by RSA RADIUS.
AMBA Custom Application Guide	Describes how the RSA Authentication Manager Bulk Administration (AMBA) command-line utility simplifies the bulk administration of users, tokens, agents, and so on. Requires a standalone AMBA license or an Enterprise license.
Online Help	
RSA Authentication Manager Help	Instructions for performing daily administration tasks in the Security Console and configuration and setup tasks in the Operations Console. Includes instructions for the most common tasks for Help Desk Administrators.
RSA Authentication Manager SNMP Help	Instructions for configuring Simple Network Management Protocol (SNMP) to monitor Authentication Manager on a hardware appliance or a virtual appliance. This Help is published on RSA Link.
RSA Authentication Manager Troubleshooting Help	The common error messages in Authentication Manager and appropriate actions for troubleshooting. This Help is published on RSA Link.
Self-Service Console Help	Describes how to use the Self-Service Console. To view the Help, on the **Help** tab in the Self-Service Console, click **Self-Service Console Help**.
RSA Token Management Snap-In for the Microsoft Management Console Help	Describes how to use software that works with the Microsoft Management Console (MMC) for deployments that have an Active Directory identity source. Using this snap-in, you can enable or disable a token, assign a token, or perform other token-related tasks without logging on to the Security Console.

Support and Service

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

The RSA Ready Partner Program website at www.rsaready.com 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.

Before You Call Customer Support

Please have the following information available when you call:

❑ Access to the RSA Authentication Manager appliance.

❑ Your license serial number. To locate the license serial 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**.

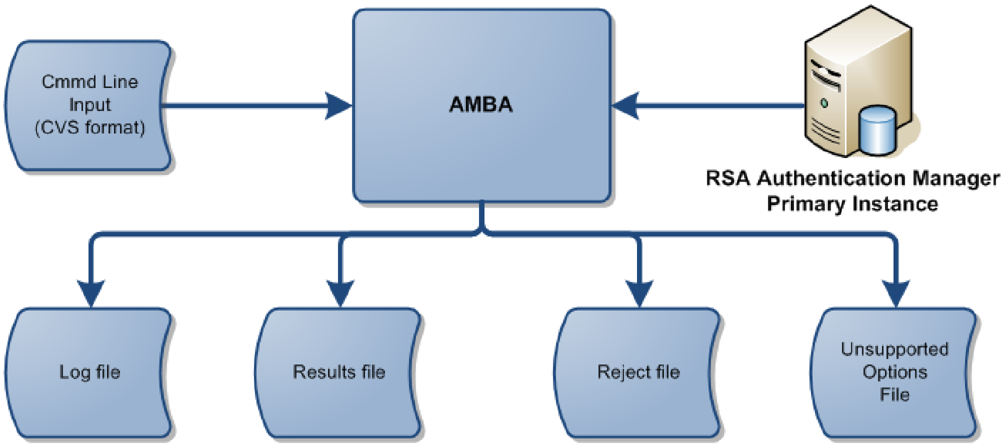
❑ The Authentication Manager appliance software version information. You can find this information in the top, right corner of the Quick Setup, or in the Security Console. Log on to the Security Console, and click **Software Version Information**.

1 Overview

Introduction

RSA Authentication Manager Bulk Administration (AMBA) supplements administrative features of RSA Authentication Manager 8.2 Service Pack 1 (SP1). This command line utility (CLU) enables Authentication Manager administrators to perform bulk administration functions from the command line interface or in a background mode through scheduled scripting.

The following diagram provides an overview of AMBA.



The AMBA utility was developed from RSA Authentication Manager Server Admin APIs. AMBA implements a sub-set of common functions available through the RSA Authentication Manager administrative consoles, including the ability to do the following:

- Perform ADD, CHANGE and DELETE operations using data from a flat (CSV, Comma Separated Variable) input file.
- LIST tokens and users that are selected based on specified criteria.
- Perform MULTIPLE token assignments, replacements, deployments and disablements.

AMBA Utility Requirements

You must meet the following requirements to use the AMBA utility.

Requirement	Description
RSA Authentication Manager Patches	Apply the most recent patches for RSA Authentication Manager. For more information, see RSA Link at **https://community.rsa.com/community/products/securid**
Operating System Support	AMBA supports all operating systems supported by RSA Authentication Manager 8.2 SP1.
Licensing requirements	AMBA requires a valid RSA Authentication Manager Enterprise license file or a standalone AMBA license.
Permissions	Super Admin credentials are required to run AMBA as a command line utility. The **rsaadmin** account password is required to log on to the appliance operating system with SSH. An Operations Console administrator account is required to enable SSH and if you need to clear the cache for troubleshooting purposes.

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

Install AMBA

To install the AMBA command line utility, place the **AMBulkAdmin.jar** file in the **/opt/rsa/am/utils/lib** directory on the RSA Authentication Manager primary instance.

Before You Begin

- You must be a Super Admin.
- Obtain the **rsaadmin** operating system password for the primary instance.
- Secure shell (SSH) must be enabled for RSA Authentication Manager. To enable SSH, an Operations Console administrator must log on to the Operations Console, and click **Administration** > **Operating System Access**.

Procedure

1. Launch the SSH client, and connect to the primary instance using the IP address or fully qualified hostname.
2. When prompted, type the operating system User ID, **rsaadmin**, and press ENTER.

3. When prompted, type the password for the **rsaadmin** operating system account, and press ENTER.
4. Change directories to **/opt/rsa/am/utils/lib**. Type:

   ```
   cd /opt/rsa/am/utils/lib
   ```

   and press ENTER.
5. Copy **AMBulkAdmin.jar** into the **/opt/rsa/am/utils/lib** directory. For example, use Secure FTP.
6. Grant the following permissions:

   ```
   chmod 600 /opt/rsa/am/utils/lib/AMBulkAdmin.jar
   ```
7. AMBA requires a valid Enterprise license file or a standalone AMBA license. Do the following:
   - If you have an Enterprise license, use the **--lic** command line option to provide the license file to AMBA. Type the following, and press ENTER:

     ```
     rsautil AMBulkAdmin --lic <license_location>
     ```
   - If you have a standalone AMBA license, copy **AMBAlicense.dat** into the **/opt/rsa/am/utils/lib** directory.

   If the command line license option is not provided, AMBA looks for **AMBAlicense.dat** in the current folder.
8. To verify that you have AMBA version 1.6.0, type the following command:

   ```
   /opt/rsa/am/utils/rsautil AMBulkAdmin -v
   ```

   and press ENTER.

   You should see the following:

   ```
   RSA AMBulkAdmin Version: 1.6.0 Build 107
   ```

Run AMBA

Use the rsautil command to execute the AMBA command line utility (CLU).

Before You Begin

- Verify that RSA Authentication Manager is running. For example, you can confirm that Authentication Manager is available by logging on to the Security Console.
- You must be a Super Admin.
- Obtain the **rsaadmin** operating system password for the primary instance.
- Secure shell (SSH) must be enabled for RSA Authentication Manager. To enable SSH, an Operations Console administrator must log on to the Operations Console, and click **Administration** > **Operating System Access**.

Procedure

1. Launch the SSH client, and connect to the primary instance using the IP address or fully qualified hostname.
2. When prompted, type the operating system User ID, **rsaadmin**, and press ENTER.
3. When prompted, type the password for the **rsaadmin** operating system account, and press ENTER.
4. Change directories to **/opt/rsa/am/utils/lib**. Type:

   ```
   cd /opt/rsa/am/utils/lib
   ```

   and press ENTER.
5. Run AMBA. Type:

   ```
   rsautil AMBulkAdmin options
   ```

   where *options* are the parameters listed in the Command-line options.

   Note: RSA Authentication Manager uses a Linux Operating System. All commands are case sensitive.

6. When prompted, enter your Super Admin User ID, and press ENTER.
7. When prompted, enter your Super Admin password, and press ENTER.
8. When are done using AMBA, close the SSH client. Type:

   ```
   exit
   ```

   and press ENTER.

Command-line options

The order of the parameters is not critical. One or more spaces are required, between the parameter key and its value.

Usage:

```
rsautil AMBulkAdmin
```

or

```
rsautil AMBulkAdmin -v
```

or

```
rsautil AMBulkAdmin --gta <tempname>
```

or

```
rsautil AMBulkAdmin --gtc <tempname>
```

or

```
rsautil AMBulkAdmin --ini <inifile>
```

or

```
rsautil AMBulkAdmin [--debug] [-g | --ctkip] [--gdir
<directory>][-i <datafile>] [-m <value>] [--newlog]
[--nolog] [-o <results file>] [-p <value>] [-r <results
```

```
file>] [--rej <command reject file>] [--verbose] [-x
<value>] [--userPwd <user password>] [-a  <value>] [-P
<value>] [--lic <license file>]
```

[…]denotes an optional parameter.

<…>denotes a value.

|denotes a choice

The characters “**[**“, “**]**”, “<”, & “>” are only used to clarify the usage of the parameters and should not be included in the actual data.

<null>

Executing **AMBulkAdmin** with a null parameter list will display a usage report on stderr.

--ctkip

Turns on the option to generate CT-KIP credentials for tokens assigned during AMBulkAdmin processing. This option is similar to the **-g option** (below), but does not produce any files. Instead, a CT-KIP activation code and download URL are generated. This option and the **-g** option are mutually exclusive.

--debug

Setting this option disables all calls to the Authentication Manager Server API and forces a successful return result. This allows processing an input file without making any changes to the database or testing various features when a database is not present. The debug option allows validating an input file for required fields without making changes to the database. The debug 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.

--datefmt <Java SimpleDateFormat>

This option overrides the formatting applied to List elements displaying a Java Date object. The default format is: E MMM dd HH:dd:ss z yyyy which would display a date as “Mon Mar 07 23:18:03 EST 2011.” For more information, see Change Date Format on page 125.

Note: Command line variables containing spaces must be enclosed in double quotes. For example, “E MMM dd HH:dd:ss z yyyy.” The same rule applies to **datefmt** statements in AMBA INI files. When **datefmt** statements with embedded spaces are included in an INI file, the number of spaces between two items is not maintained. If this is an issue, include the **datefmt** statement on the command line.

-g

Turns on the option to output RSA 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 are given the extension **.sdtid**. For example, if the default login name is *juser* and if the token serial number being assigned is *27050105*, then the filename will be *juser_000027050105.stdid*. By default, files are stored in the current directory, but this may be changed using the **-gdir** option. This option and the **--ctkip** option are mutually exclusive.

--gdir <dirname>

Specifies that SecurID Software Token database files should be stored in the directory specified by **<dirname>**. The **-g** option must also be specified to cause the database files to be saved.

--gtc <tempname>

Generate a CSV template file. This file contains a header line and a line with the correct number of empty columns. One or more comment lines are also emitted. This file can be used as the base for developing a CSV input file. <**tempname>** is a fully qualified path and file name that is used for this file. **<tempname>** must be fewer than 128 characters.

-i <datafile> | stdin

Where **<datafile>** is the path to the CSV formatted input file, the pathname must be fewer than 128 characters. The literal **stdin** may be supplied to redirect system standard input into AMBulkAdmin. The **quit** command is used to terminate a standard input file.

--ini <inifile>

Where **<inifile>** is the path to the input parameter file. The pathname must be fewer than 128 characters. This is a text file containing input parameters and is formatted as described below. Additional command line parameters will have precedence over duplicate ini file parameters.

-m <0 | 1 | 2 | 3>

The default messaging level is 0, log all messages. Levels 1, 2 and 3 all log BOJ, EOJ and application error messages and do not log information messages. In addition, level 1 logs successful command messages, level 2 logs failed command messages and level 3 logs successful and failed command messages.

Msg Type / Level	0	1	2	3
Boj	Yes	Yes	Yes	Yes
Eoj	Yes	Yes	Yes	Yes
Info	Yes	No	No	No
Error	Yes	Yes	Yes	Yes
Successful	Yes	Yes	No	Yes
Failure	Yes	No	Yes	Yes

Here is an example of each message type:

```
BOJ    : 2009-02-16 17:35:13 - RSA AMBulkAdmin version 1.0; 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    : 2009-02-16 17:35:18 - Terminating
```

--newlog

This option forces AMBA to create a new log as opposed to the default option that appends new log information to any previous log information.

--nolog

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

-o <log file> | stderr | stdout

Where **<log file>** is the path to a file for storing the log output. If **<log file>** is not specified, the message information is printed to the standard output channel, where it may be redirected using standard operating system conventions. The pathname must be fewer than 128 characters. The output filename is optional, if not specified the default is **AMBulkAdminlog.txt** and will be placed in the current directory. Either the literal **stderr** or **stdout** may be used to redirect log output to system standard error or standard output files.

-p <1 | 2 | . . . 3600>

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

-r <results file> | stdout | stderr

Where **<results file>** is the path and file name for storing the results of a **List** command. This file is overwritten on each execution of AMBA. If this option is not specified a default file name of **AMBulkAdminResults.CSV** and will be created in the current directory. Either the literal **stderr** or **stdout** may be used to redirect log output to system standard error or standard output files.

--rej <command reject file>

Where **<command reject file>** is a fully qualified path and file name to be used for this file and must be fewer than 128 characters. This is a file containing rejected input records in the same format as the input file. If this option is not specified a default file named **AMBulkAdminReject** will be created in the current directory. If the default file name is used, the file extension of the command input file will be appended to it.

--searchlimit

Searchlimit determines the maximum number of principal objects that will be 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 **searchlimit** to something slightly larger (**+100**) then the number of principals in the database. Failure to do so will truncate the results of various commands.

-v

Returns the AMBA version number.

--verbose

Enable enhanced logging. This function is usually only for debugging. It mainly generates Information message types and records program flow. Using this option can severely degrade program performance.

-x <0 | 1>

Defines if a datestamp is inserted into the name of the following files: **log** file (**-o**), **reject** file (**--rej**), **results** file (**-r**) and the software token database files (**--gdir**). If not specified, default is **0**.

--userPwd <user password>

Where **<user password>** is a password value that meets the password policy requirement defined. In Authentication Manager, users are assigned with password. While “adding a user” or “changing a user,” this option can be used to have all the users get the same password. Without this option, a password for each user can be provided in the input file under **UserPwd** field.

Note: Exercise caution when using this command line option. If the input file contains multiple actions (for example, **addUser** and **ChangeUser**), then the value of **<user password>** will be used for both the, add and change options. Ensure that the input file data is appropriate before using this option.

-a <value>

Super admin user id

-P <value>

Super Admin password

--lic <license file>

Where **<license file>** is the location of the AMBA license file. If this command line option is not specified, AMBA will look for **AMBAlicense.dat** file in the current directory.

Note: The bash shell on the appliance interprets the ‘**!**’ character as a special character referencing the shell's command buffer. Other UNIX/LINUX shells may 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 on page 31.

2 Input File Processing

Implementation

The input file will be processed in sequential order. It is up to the RSA Authentication Manager administrator to arrange transactions in a logical order. 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. No attempt will be made to add the group from the add user command.

Each command will consist of an action code, required fields and zero or more optional fields. Commands consist of two or more character codes. For a command to succeed, all required fields must be present, although it may still fail for other reasons.

Some optional fields may cause a subordinate command to be executed. Failure of the subordinate command will not cause the primary command to fail. An example of this is adding a user and a group association. If the users add is successful, then a group association will be attempted. If the group association fails it will be logged, but the add user still stands. For example, in the following input file, we are trying to add William Doe as a new user. In addition we would like to make William Doe a member of BigGroup6.

Input:

```
Action,LastName,FirstName,DefLogin,,,,,,,GrpName
au,Doe,William,CIC,,,,,,,BigGroup6
```

Examining the log, we see that the primary command, adding William Doe as a new user was successful; however, the secondary command of making William Doe a member of BigGroup6 failed because there is no such group. If we examine the database, we will find that William Doe is now a valid user, but he is not a member of BigGroup6. In other words, the primary command still stands, but the secondary command does not.

```
BOJ    : 2009-02-16 18:18:44 - RSA AMBulkAdmin version 1.0; Input =
f:\input.csv
Info   :                                        -Output Log File Opened

Info   : Line    1                               -Header Line

Info   :                                         -Entering addUser

Success: Line    2 - addUser                      -user2, user2

Info   :                                        -Adding User to Group

Failure: Line     2 - addUser to Group          -user2,Group1 API
return:Invalid Group or Principal found
```

```
Info   :                                  -Closing input file

Info   :                                      -Closing rejected actions
file
Info   :                                          -Closing unsupported
actions file
Info   :                                  -Log File Closed

EOJ    : 2009-02-16 18:18:50 - Terminating
```

Examining the command reject file, we see the original input line and a comment specifying that the group was not found.

Command Reject file:

```
action,DefLogin,LastName,FirstName,SecurityDomain,IdentitySo
urce,GrpName
// Line 2:  Invalid Group or Principal found
au,user2,user2,,,,Group1
```

To correct the error, change the **AU** action to **AG** in order to add the group. Then duplicate the line and change the **AG** to **AUG** in order to add the user to the group. Submit this file as input to the next AMBA run.

Corrected input file:

```
action,DefLogin,LastName,FirstName,SecurityDomain,IdentitySo
urce,GrpName
// Line 2:  Invalid Group or Principal found
ag,user2,user2,,,,Group1
aug,user2,user2,,,,Group1
```

The AG command will ignore the unused fields (**LastName, FirstName** and **DefLogin**) and create a group titled **Group1**. The **AUG** command will add **user2** to **Group1** because **user2** is a valid user (from the earlier run) and **Group1** is now a valid group from this run.

Note: Minor editing was performed on the examples above to facilitate fitting them into this document.

For a table of supported commands, see Appendix C, Command Table.

Preparing the Datafile

The data file should be in CSV (Comma Separated Variable) format. In CSV format, a comma separates each data field with no double quotes (“).

Important: A comma is an illegal character within a field.

The best environment to create the data file is a spreadsheet or word processor application that is capable of saving in CSV format, such as Microsoft Excel.

It is not necessary for input lines to contain empty fields beyond the last significant data entry field. For example, **addGroup** (**AG**) need only have all fields through the **GrpNam**e field, whereas **addUser** (**AU**) would contain all fields. The unused fields would be delimited by ,, (comma comma) for CSV files.

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

Note: Do not to leave the first line in the CSV file as empty or blank. The utility will ignore the blank line and will not consider the first line as line number 1.

```
1
2 Action,IdentitySource,SecurityDomain,LastName,FirstName,
DefLogin,...
3 <data>,<data>,<data>,<data>,<data>,<data>,<data..
4 // Replace <data> with actual data or delete it leaving ""
then delete this comment line.
5 // If desired the header labels or first line of this file may
also be deleted.
```

In this example, line 2 would be considered line 1, which would be misleading while reading the log files.

Header lines

A header is used to indicate what data is present and what order it will appear in the input file. Header lines are optional, however if a header is present, it must be the first line of input and its first field must be **Action** or **"Action"**. All other fields and their position are optional.

Available field names:

Action, IdentitySource, SecurityDomain, LastName, FirstName, DefLogin, DefShell, UserPwd, TokSerial, ReplTokSerial, TokEnabled, SetPin, CreatePin, PinMode, Pintype, GrpName, GrpDefLogin, GrpDefShell, ClntName, AgentHostName, SoftIDParams, SoftIDPW, RemoteAlias, RealmName, CompareField, CompareType, CompareValue, OutputOption, ExtnDataOption, MiscVariable, ProfileName, TokenSerial, rangeMode, startRange, endRange, Password, Filename, copyProtect, overOption, Email, CertDN, Key, KeyType, SecurityDomainName, ParentDomainName, SecurityDomainDescription, SecurityDomainCreatedBy, PolicyType1, PolicyName1, PolicyType2, PolicyName2, PolicyType3, PolicyName3, PolicyType4, PolicyName4, PolicyType5, PolicyName5, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, DestinationSecurityDomain, SubDomain, Limit, MinTokenLIfe, RegenerateSeed, OTPLength, OTPInterval, OTPAlgorithm, PinAdded, NicknameIsCtkipCode, DeviceserialIsCtkipCode, PINIndicator, InstanceName, ExpiryDate, DeliveryMethod, DestinationAddress, EnableFlag, ForceGroupSearch, SubCommand, AttributeName, AttributeValue

The header line is not case sensitive.

Because a header line can only appear as the first line of input, it is in effect for the entire input file, unless it is replaced by a new header through the **CIF** (**Change Input Format**) command. For more information, see Change Input Format on page 126.

The **CIF** command may be issued in any position of the input file and may appear any number of times. Its only prerequisite is that it must always contain an action field. Any additional parameters must be picked from the available field names listed above.

The action field, along with any other fields may 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 a **CIF** command positions the action field in other than the first field, then a subsequent **CIF** command would have to be carefully formatted in order to ensure that the **CIF** action appears in the correct column. A **CIF** command is simply a header preceded by the command "**CIF**." An example of a **CIF** command is:

cif,action,grpname

This would imply that all input following this CIF action would consist only of an action code followed by a **grpname**. This example would be useful if a large number of groups are to be created. It would only be necessary to provide the command, such as **AG** and the **groupname** for each group to be created.

Each function is explained in detail in this guide. When you run a command, the fields fall into two categories:

Required

The function cannot be performed without this data and will return an error if missing.

Optional

The function will use the data, if provided.

Field Definitions

These definitions are general in nature; however, specific commands may 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 for PIN numbers for related commands, it is also used to supply passwords for password-related commands.

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

Action	Description
AttributeName	Used to provide the name of a custom attribute.
AttributeValue	Used to provide a value for a custom attribute named in attributeName.
CertDN	User’s certificate DN.
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. The reason for this is to keep AMBA aligned as closely as possible with ACEBulkAdmin.
CompareField	Used in the list commands to indicate which field to use as a filter and selector for report data selection. If this field is zero, or empty, the **CompareType** field is assumed to also be zero or empty. Consult the individual list command definitions for allowable entries for this field.
CompareType	Used in the list commands to indicate what type of comparison to apply to the **CompareField**. If this field is zero, or empty, the **CompareField** field is assumed to also be zero or empty. Consult the individual list command definitions for allowable entries for this field.
CompareValue	This field is used to supply values for list command filter/selector report data selection. Consult the individual list command definitions for allowable entries for this field.

Action	Description
CreatePin	This field is used to set the validity period of emergency access codes. The format is "DnHn", for example, "D90H0" = 90 days, "D0H12" = 12 hours. If not specified, 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. Optionally, an **L** component may also be present (**DnHnLn**). The **L** component 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.
DefLogin	User's default login or account name, used when assigning a user to a Group where a Group-specific login name is not supplied. The maximum is 255 characters.
DefShell	User's default shell, used when assigning a user to a Group where the Group-specific shell is not supplied. The maximum is 256 characters.
DeliveryMethod	Designates types of notification (SMTP or SMS), and in some cases activates the notification.
DestinationAddress	Used for email addresses and SMS phone numbers. For email addresses, it will override any existing principal email address.
DestinationSecurityDomain	Used to declare the destination security domain for the **CUSD** command.
DeviceSerialNumber	Used to provide a device serial number for software token activations.
DeviceserialIsCtkipCode	Used to indicate the **Deviceserial** field value will be used for the CTKIP Activation Code. This is only applicable when the **--CTKIP** command line option is declared.
Email	User's email address
EnableFlag	True to enable a user account, false to disable the account.
ExtnDataOption	This field is used to indicate whether or not user extension data should be included in the list. Consult the individual list command definitions for allowable entries for this field
FirstName	User's real first name to a maximum of 255 characters.
ForceGroupSearch	A user may only be added to a group in the same **IdentitySource** or the **Internal Database**. If this variable is set **true**, a **groupName** will be searched for in the Internal Database. Any other value (including empty or missing) will cause the group to be searched for in the user's **Identity Source**.

Action	Description
GrpDefLogin	User’s login or account name for Agents that the specified Group is activated on, maximum 48 characters. If not supplied, the **DefLogin** value will be used.
GrpDefShell	User’s shell for Agents that the specified Group is activated on, maximum 256 characters. If not supplied, the **DefShell** value will be used.
GrpName	Name of the Group in which to register the user. The maximum value is 255 characters.
IdentitySource	Identity Source in which the search needs to be done.
InstanceName	Provides the name of the target instance when multiple instances have been configured.
Key	Used for search and lookup arguments.
KeyType	The type of search or lookup argument supplied in Key.
LastName	User’s real Surname or Family name, to a maximum of 255 characters.
Limit	Used for various limits for some commands.
MiscVariable	This field is used in various commands and is used to supply miscellaneous information to the command. The definition of the contents of this field can be found in the specific command descriptions where it is declared.
Nickname	Used by token assignment commands to supply a value to the token nickname when applicable.
NicknameIsCtkipCode	Used to indicate the **Nickname** field value will be used for the CTKIP activation code. This is only applicable when the **--CTKIP** command line option is declared.
OTPAlgorithm	Used to configure a soft token as either a time-based or event-based token.
OTPInterval	Used to configure a soft token display duration as either 30 or 60 seconds.
OTPLength	Used to configure a soft token display as either 6 or 8 characters in length.
OutputOption	This field is used to apply formatting and other options to the list commands. It declares whether a header line should be output to the list, and whether extended user fields should be output to the list. Consult the individual list command definitions for allowable entries for this field.

Action	Description
ParentDomain	Name of parent security domain for security domain to be added or deleted.
PinAdded	Used to configure a soft token as either passcode (PIN added without carry to tokencode) or tokencode (PIN pre-appended to tokencode).
PinMode	Used to indicate whether or not new PIN mode should be set. **Note:** Resetting **pinMode (value=0)** will only have an effect if the token already has a pin.
PinType	Used to indicate a passcode token (PIN + tokencode) or a tokencode token (no PIN).
PolicyName1 thru 5	Used to provide the policy name to be selected for the matching **policyType** in the **ASD** command.
PolicyType1 thru 5.	Used to specify the **policyType** for policy selection in the **ASD** command. **Note:** Each **PolicyType** must have a matching **policyName** entry. Valid entries are as follows: **PasswordPolicy** **LockoutPolicy** **SelfServicePolicyAM_Token_Policy** **AM_OFFLINE_AUTHN_POLICY** Entries are case-sensitive.
ProfileName	Used to provide a profile name during user profile maintenance.
RealmName	Used to provide a realm name for remote user maintenance.
RegenerateSeed	Used to force or inhibit the generation of new tokencodes during soft token deployment.
ReplTokSerial	Replacement Token serial number, up to 12 numeric characters (0-9). Leading zeros are optional. If only the Token Serial number is specified, then the Token will be replaced immediately.
SecurityDomain	Security Domain in which the search needs to be done.
SecurityDomainName	Name of security domain to add or delete.
SecurityDomainDescription	Description of Security Domain to be added.
SecurityDomainCreatedBy	**CreatedBy** entry for added security domain

Action	Description
SetPin	Sets the PIN status: **C**, **c** or **0** = New PIN Mode, PIN Cleared. **N** = New PIN Mode, Old PIN Required, not valid in Action = **Add** mode. Any other string value will attempt to set the PIN to that value. System PIN rules determine the success of setting the PIN. This field is also used to supply a password for the **Add User and Password** (**AUP**) and **Change User and Password** (**CAUP**) commands. For more information, see Add User and Password on page 44 and Change or Add User and Password on page 62. For the emergency access commands, this field is used to supply a password, or it can be used with the following format to set the number of passwords to generate. **Nn** n - Number of one time passwords (OTP) to issue. Format of **SetPin** in AceBulkAdmin was **NnLnFn**. However, in RSA Authentication Manager, the Length and Format of one-time passwords are set by token policies and cannot be set manually through a command. Any values that are given for the Length and Format fields are ignored. Check the token policies in Authentication Manager before using this field.
SiteFile	Used to provide the URL for the software token site url LIST during token activation
SiteURL1	Used to provide URL1 for the software token site1 during token activation.
SiteURL2	Used to provide URL2 for the software token site1 during token activation.
SiteURL3	Used to provide URL3 for the software token site1 during token activation.

Action	Description
SoftIDParams	This is a three digit (to maintain consistency with ACEBulkAdmin utility) field used by the add-token commands when a softID token seed record file is being created. When used, this field must contain three decimal digits that control the following seed file generation characteristics: First digit: 0 - required but ignored Copy Protection Flag 0 or 1: 0-Copy protection off 1 -Copy protection on Password usage and Interpretation method 0, 1, 2, or 3: 0- No password 1- Static password (See SoftIDPW below) 2- Default login 3- Default login appended to static password
SoftIDPW	This field is used by the add-token commands when a softID token seed record field is being created. This field supplies a password to be used for the seed file encryption when **SoftIDParams** specifies a static password.
SubDomain	Used to indicate whether or not a security domain search should include sub-domains.
TemplateFile	The XML template file to use for constructing email notifications. If empty, AMBA will use a built-in default template. There are default template files included with the AMBA distribution that may be used as a starting point for building your own.
TokEnabled	Post-assignment Token status, single character, ‘0’ (zero) = disabled or ‘1’ (one) = enabled. If not specified will default to ‘0’ (disabled). This action only affects the specified Token.
TokSerial	Token serial number, up to 12 numeric characters (0-9), leading zeros are optional.
UserPwd	IMS Password for each user. Instead of providing a password for each user, password can be provided using “--userPwd” command line option. This field takes precedence over the command line option.

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

User Information Fields

chLastName	User’s last name
chFirstName	User’s first name
chDefaultLogin	User’s default login
chDefaultShell	User’s default shell
bTempUser	Whether user is a temporary user (TRUE/FALSE)
dateStart	Start date for temporary user (stored as Coordinated Universal Time, or UTC, time)
dateEnd	End date for temporary user (stored as Coordinated Universal Time, or UTC, time)

Token Information Fields

chSerialNum	Token serial number
iInterval	Number of seconds between display changes
dateBirth	Date the token was activated
dateDeath	Date the token will shut down
dateLastLogin	Date of the last login with this token
iType	Token type: 0 RSA SecurID Standard Card 1 RSA SecurID PINPAD Card 2 RSA SecurID Key Fob 3 RSA SecurID Watch 4 RSA SecurID Software Token (formerly SoftID) 5 RSA SecurID Smartcard 6 RSA SecurID Modem 7 RSA SecurID Crypto 8 RSA SecurID Proteus 9 RSA SecurID USBCOSMO (SID800) 10 RSA SecurID Flextoken
bHex	Whether the display is hexadecimal (TRUE/FALSE)
bEnabled	Whether the token is enabled (TRUE/FALSE)

bNewPINMode	Whether the token is in New PIN mode (TRUE/FALSE)
bMustCreatePIN	Whether user must create PIN (TRUE/FALSE)
iNextCodeStatus	Next token code status: 0 Not in next token code mode 1 Token is in next token code mode
iBadTokenCodes	Number of bad token codes 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 RSA 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	Whether software token is currently deployed: 0 No 1 Yes
iCount	Number of times token has been deployed
ExtnKey	Attribute Name of User/Token being requested.
ExtnData	Attribute Value of User/Token being requested.

Software Token Device Type Fields

chSTDTDescription	STDT Description field.
chSTDTFamilyKey	The family key could 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	STDT Label Key

chSTDTPluginModuleName	The plug-in module name is used for token exporting. This could be one of two system-embedded plug-ins: PC Software Token V2.4 or earlier for 64-bit tokens and PC Software Token V3.0 or later for 128-bit tokens, or it could be customized plug-ins, such as the TSF plug-in.
chSTDTVersion	STDT Version.
bSTDTIsPinpad	STDT Pin type is pinpad, true or false.
chSTDTTokenCodeLength	STDT tokencode length, 6, 8, 6|8, 8|6
chSTDTTokenCodeInterval	STDT tokencode interval 30, 60, 30|60, 60|30
chSTDTTokenCodeType	STDT tokencode type, time or event

Input Parameter File

Input parameter files may be created and used in place of command line arguments. The file is a text file, consisting of command line parameters. One or more spaces must separate the command line parameter and any arguments.

rsautil AMBulkAdmin --ini example.ini

Listing of one 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
```

As of AMBA version 1.3, when an INI file is first encountered, it is automatically encrypted, replacing the unencrypted version. From that point on, the encrypted version will be used. If the unencrypted version cannot be encrypted and replaced, the process will fail and AMBA will terminate with an error.

Note: There is no process to unencrypt an encrypted INI file.

If an unencrypted version is needed, it is up to the user to make appropriate backup copies, prior to using an unencrypted INI file, with AMBA.

Note: The bash shell on the appliance interprets the ‘!’ character as a special character, referencing the shell's command buffer. One work-around for this is to use the INI switch and put these parameters into an INI file. Use of an .ini file will also prevent a command-line password from being captured in the Linux history file.

Command Reject File

Each run of AMBA will produce a command reject file. Its name and location are configurable through command line entries.

This file will contain a copy of each input line that fails for any reason. A comment line describing the reason the command failed precedes each entry in the command reject file and includes the line number of the associated input line. If a primary command succeeds and the secondary command fails, the line will be entered into this file.

The purpose of this file is to provide a convenient way of correcting input errors. To correct any input errors, simply edit this file, correct any errors, and supply this file as input to the next update. In cases where a primary command succeeded and a secondary command failed, the secondary portion of the command should be corrected and converted to primary command. This is because the old primary command most likely would not succeed the second time, and the secondary command would not be attempted.

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
```

Reporting and Logging

In addition to RSA Authentication Manager logging and reporting, AMBA will produce an optional log and a standard command reject file. The command reject file is defined above, along with an example of its contents.

Logging functions are explained in Command-line options on page 14. The “--verbose” logging function is mainly used for debugging problems. It mainly issues “Info” type log messages.

The “Boj” and “Eoj” type log messages should need no explanation. The “errMsg” type is caused by application type errors, such as invalid file names and directories and command line errors. “Success” and “Failure” are reserved for easy identification of the final result of a command or a secondary command. If “API Return: appears in a “Failure” type message, it indicates that the Authentication Manager API returned and error and its text follow this string. All other “Failure” type messages are returned by AMBA. “Success” and “Failure” message type also contain the line number of the associated input line.

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

```
BOJ    : 2009-02-13 18:15:08 - RSA AMBulkAdmin version 1.0; Input =
f:\input.csv
Info   :                                 -Output Log File Opened

Info   : Line    1                        -Header Line

Info   :                                  -Entering addUser

Success: Line    2 - addUser               -user1, user1

Info   :                                  -Leaving addUser

Info   :                                 -Entering addTokenToUser

Failure: Line     3 - addTokenToUser             -null, user1 1 API
return: zero (0) tokens found matching serial number:000123456789
Info   :                                  -Closing input file

Info   :                                     -Closing rejected actions
file
Info   :                                         -Closing unsupported
actions file
Info   :                                  -Log File Closed

EOJ    : 2009-02-13 18:15:13 - Terminating
```

Default File Names

If file names are not specified either by command line arguments or input parameter file statements, the following default filenames will be used:

Log filename:

AMBulkAdminLog.txt

Input reject filename:

AMBulkAdminReject (Extension determined by input file extension.)

Results filename:

AMBulkAdminResults.CSV

If default file names are used, then paths (directories) have not been provided. In this instance, output files will be created in the current directory.

Input Template Files

Input template files can be created to assist in creating input files (See the **--gta** and **--gtc** commands). A template file may be used as a base file for creating AMBA input files. The template file creation option is mutually exclusive of other command line options. If this option is specified, any additional options will be ignored.

The following is an example of a CSV template:

```
Action,IdentitySource,SecurityDomain,LastName,FirstName,DefLogin,DefShell...
<data>,<data>,<data>,<data>,<data>,<data>,<data...
// Replace <data> with actual data or delete it leaving "" then delete this comment
line
// If desired the header labels or first line of this file may also be deleted.
```

Note: The header and data lines of the above example files have been truncated in order to fit into this format and to avoid line wrap.

Sample CSV Format Data

(1 line header, 1 line data; no embedded ‘cr/lf’)

Action,LastName,FirstName,DefLogin,DefShell,TokSerial,ReplTokSerial,TokEnabled,SetPin,GrpName,GrpDefLogin,GrpDefShell

AUT,Smith,John,Smithj,,853618,,1,1234,local,,,fred.securid.com

AMBA Return Values

AMBA will return **0** if no errors are detected, otherwise AMBA will return a number greater than **0**. There are two types of errors that may be detected and reported. The first type will produce an “Error” message in the output log and is usually the result of a sub-command failure. This usually but not always will result in a command failure. There may be multiple “Error” messages for a single command. For one or more “Error” messages, a value of **1** will be returned. A command failure will produce a “Failure” message in the output log. For one or more “Failure” messages, a value of **2** will be returned. If both “Error” and “Failure” messages have been detected, then a value of **3** will be returned. 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 will be sent to stdout)

Note: AMBA is run by rsautil; therefore, any return value is intercepted by rsautil. To have rsautil return this value to a calling task such as a script, use either the **-S** or **--script-exit** rsautil parameter. For example:

```
rsautil --script-exit AMBulkAdmin <amba parameter list>
```

This causes rsautil to pass any AMBA return result back to a calling process.

3 Add Functions

Add User

Add a new user and optionally add a user to an existing group.

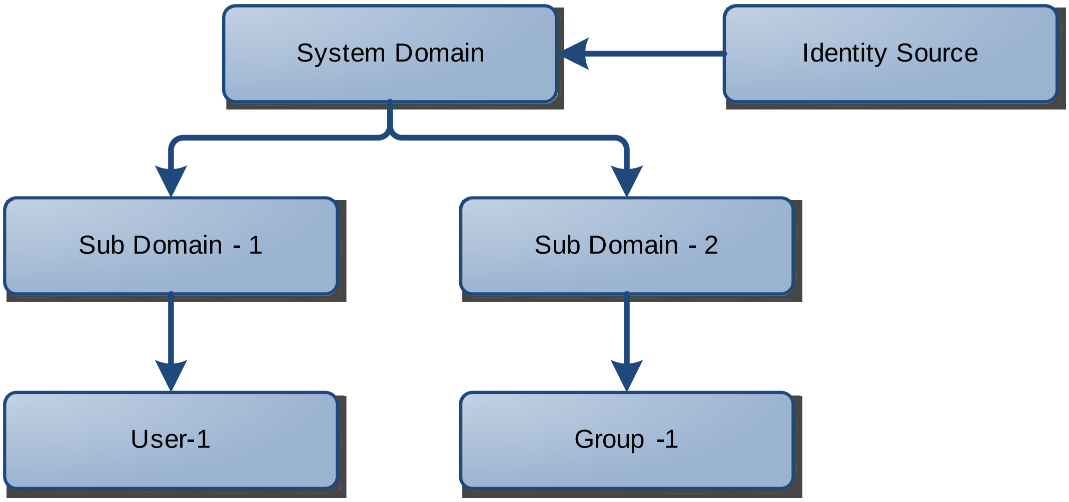
Action	**AU**
Required Fields	LastName, DefLogin
Optional Fields	FirstName, Email, CertDN, DefShell, GrpName, GrpDefLogin, GrpDefShell, UserPwd, IdentitySource, SecurityDomain, EnableFlag, ForceGroupSearch, AttributeName, AttributeValue, AttributeName1, AttributeValue1, AttributeName2, AttributeValue2,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

If Security Domain and Identity Source are not provided, then the user will be added in the default identity source (Internal Database) and security domain (System Domain) created during Authentication Manager installation. Group does not need to be present in the same domain as the user, but the group should be under the same realm. This command finds the group by using the user’s security domain’s parent domain. A user may only be added to a group in the same Identity Source or the Internal Database.

If **ForceGroupSearch** is set **true**, a **groupName** will be searched for in the Internal Database. Any other value (including empty or missing) will cause the group to be searched for in the user's Identity Source.

AttributeName and **AttributeValue** are numbered pairs. They may be used to provide a value for a custom attribute named under the respective attribute name pair. The number pairs may be declared in any order, and the attribute name will be searched in the user’s **IdentitySource** (it will not create or define new attributes).

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 assigned is mentioned: **Sub Domain-1**. This command will internally process to find **Group - 1**, by identifying the parent domain of **User - 1**, which in this case is **System Domain**, and then use this as the base domain to search for **Group-1**.

Note: User Password is required to add a user. Either each user can be provided with different password using **UserPwd** optional field or the cmd line option **--userpwd** can be used to have the same password given for all the users. If Identity Source is **Internal Database** and if no password is provided in both the places, Users will be added with password as null. If Identity Source is an External Directory like an **LDAP server**, the command will throw an exception.

Add User and Token

Adds a user and assigns the token specified by **TokSerial**. The token is enabled, the PIN is cleared, and both **BadTokenCodes** and **BadPINs** are set to zero. The **TokEnabled** field defines the token state following successful assignment. Additionally, the PIN can be set or cleared depending on the value of the **SetPin** field. If the user has been added previously a FAILURE message will be generated. The user may also be assigned to an existing Group. For additional information on field values, see Field Definitions on page 23.

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, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, RegenerateSeed, OTPLength, OTPAlgorithm, OTPInterval, PinAdded, NicknameIsCtkipCode, DeviceserialIsCtkipCode, EnableFlag, ForceGroupSearch, DeliveryMethod, DestinationAddress, AttributeName, AttributeValue, AttributeName1, AttributeValue1, AttributeName2, AttributeValue2, AttributeName3, AttributeValue3, AttributeName4, AttributeValue4

If Security Domain and Identity Source are not provided, then the user will be added in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation. Token identified by the **TokSerial** and Group identified by **GrpName** do not need to be present under the same domain as the user, but should be under the same realm.

A user may only be added to a group in the same IdentitySource or the Internal Database. If **ForceGroupSearch** is set **true**, a **groupName** will be searched for in the Internal Database. Any other value (including empty or missing) will cause the user's **IdentitySource** to be searched for the group.

AttributeName and **AttributeValue** are numbered pairs. They may be used to provide a value for a custom attribute named under the respective attribute name pair. The number pairs may be declared in any order, and the attribute name will be searched in the user's IdentitySource (it will not create or define new attributes). Consult the **ctkip** command line option and the **SSTDT** command for ctkip credentials generation options.

If the **--ctkip** command line option has been declared and **NicknameIsCtkipCode** is set to **true**, the contents of the **Nickname** field will be used for the ctkip activation code. If **DeviceserialIsCtkipCode** is set to **true**, the contents of the **DeviceSerial** field will be used for the ctkip activation code. Both options set to **true** will throw an error.

Filename may be used to rename the output file. If left empty, the **fileName** defaults to **<user ID>_<token serial>.sdtid**.

There are five optional parameters available to configure Soft Token deployments:

- Set **RegenerateSeed** to **true** to force the generation of new tokencodes or **false** to leave the current tokencodes in place. The default value is **true** if missing or empty.
- Set **OTPLength** to force the tokencode length. Valid values are **6** and **8** to force tokencode length to 6 digits and 8 digits respectively. Leave this parameter empty to generate the default value for tokens being deployed.
- Set **OTPInterval** to force the tokencode display duration. Valid values are **30**, **60** and empty. 3**0** and **60** set the display duration to 30 or 60 seconds respectively. Empty will generate the default value for tokens being deployed.
- Set **OTPAlgorithm** to **Time** to configure the token as a time based token or to **Event** to configure the token as an event-based token. Setting **OTPAlgorithm** to empty will generate the default value for tokens being deployed.
- Set **PinAdded** to **Passcode** to configure a soft token PIN as a passcode (pin-pad) token. In this mode, the PIN is added to the tokencode without carry. **Set PinAdded** to **tokencode** to configure a soft token PIN as a tokencode. In this mode, the PIN must be pre-**appended** to the token. Leave this parameter empty to generate the default value for tokens being deployed.

Automatic Notification

The following fields are used for automatic notification. When provisioning software tokens, Automatic Notification may be used for delivery of the following information:

- For CT-KIP provisioning the activation code and delivery URL.
- For SDTID file provisioning the token sdtid file. Applies to both password protected files and unprotected files.
- For SDTID files that are password protected. The token file will be sent in one email and the password will be sent in a separate email.

For more information, see Appendix A, Automatic Notification.

DeliveryMethod - For email delivery set this field to one of the following values. Leave it empty to disable notification.

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

DestinationAddress - For SMTP this field can be used to provide an email address. If this field is empty or not used the principal’s account email address will be used.

TemplateFile - Use this field to provide the path and file name of an email template file. Template files are included in the AMBA package that may be used as a starting point. Providing a customized template allows additional text to be included in the email along with changes in format. If a custom template file is not provided, AMBA will use a built-in template.

InstanceName - Name of current instance. This is only required if **DeliveryMethod** is set to SMTP and multiple Instances have been declared for this server.

The **-g command** line option is required to instruct AMBA to generate software token database files. Optionally, the **-gdir** command line option may be used to place any generated files in the specified directory. The --ctkip and -g options are mutually exclusive

The **--ctkip** command line option is required to instruct AMBA to generate CT-KIP credentials for qualifying tokens. The **--ctkip** and **-g** options are mutually exclusive. Also, the set soft token device type command (**SSTDT**) can be used to force a specific device type for credential generation. For more information, see Set Soft Token Device Type on page 71.

If the **-g** and **--ctkip** options are not used, the AUT command will assign tokens but will not build any output files or generate any CT-KIP credentials. Additionally, if the **-g** command line option is not used, any **-gdir** command line option will be ignored.

Software Token Device Type Attributes

A specific Software Token Device Type may be linked to a software token through the security console, the API or the AMBA **SSTDT** command. Software Token Device Types contain various sets of attributes which are basically name/value pairs. Default values may be assigned to these attributes through the security console, the APIs or AMBA. Use the variables declared in the following table to assign values to the attributes for specific tokens.

Software Token Device Type Attributes	AMBA Variable 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	SiteRUL2	0, value, empty
TOOLBAR_SITEURL3	SiteURL3	0, value, empty

Attribute values:

-2 - copy the TokSerial to the attribute value field.

-1 - copy the DefLogin to the attribute value field

0 - force 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

Note: “User Password” is required to add user to IMS. Either each user can be provided with different password using **UserPwd** optional field or the cmd line option **--userpwd** can be used to have the same password given for all the users. If **Identity Source** is **Internal Database** and if no password is provided in both input file and as a command line option, users will be added with password as null. If Identity Source is an **External Directory** like an LDAP server, the command will throw an Exception

Add User and Token Automatic

This command automates the Add User and Token command by obtaining an unassigned token of a specified type from the system and calling the **Add User and Token** command using the newly acquired token serial number. The field definitions and requirements are identical to those of the **AUT** command except for the following two exceptions:

The **MiscVariable** field is required and is used to supply the desired token type. Acceptable values are:

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

Only **Non Expired Tokens** will be searched and assigned to users. The **MinTokenLife** field is optional and is used to guarantee that a token will have a minimum number of days before its expiry date.

The search for a token looks for tokens, with an expiry date greater than the current date. Adding a number of days with the **MinTokenLife** variable will adjust the search to tokens with an expiry date greater than **today + MinTokenLife**. For example, if **MinTokenLife** is set to **90**, the token search would only get tokens with and expiry date greater than **today + 90** days, meaning the token would not expire for at least 3 months or more. **MinTokenLife** is ignored if **MiscVariable** is set to **-1**.

The **TokenSerial** field is not required and is ignored if present.

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, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, RegenerateSeed, OTPLength, OTPAlgorithm, OTPInterval, PinAdded, NicknameIsCtkipCode, DeviceserialIsCtkipCode, EnableFlag, ForceGroupSearch, ,DeliveryMethod, DestinationAddress, TemplateFile. InstanceName, AttributeName, AttributeValue, AttributeName1, AttributeValue1, AttributeName2, AttributeValue2, AttributeName3, AttributeValue3, AttributeName4, AttributeValue4

If Security Domain and Identity Source are not provided, then the user will be added in the default identity source (Internal DataBase) and security domain (System Domain) created during RSA Authentication Manager installation.

The group identified by **GrpName** does not need to be present under the same domain as the user, but should be under the same realm. A user may only be added to a group in the same **IdentitySource** or the Internal Database.

If **ForceGroupSearch** is set **true**, a **groupName** will be searched for in the Internal Database. Any other value (including empty or missing) will cause the user's **IdentitySource** to be searched for the group.

AttributeName and **AttributeValue** are numbered pairs. They may be used to provide a value for a custom attribute named under the respective attribute name pair. The number pairs may be declared in any order, and the attribute name will be searched in the user's **IdentitySource** (it will not create or define new attributes).

Token will be searched in same realm as that of user while assigning token of a particular type. However if first available unassigned token is requested (-1) then the Authentication Manager API gets the token which expires first irrespective of Security Domain.

Consult the **ctkip** command line option and the SSTDT command for **ctkip** credentials generation options.

If the **--ctkip** command line option has been declared and **NicknameIsCtkipCode** is set to **true**, the contents of the **Nickname** field will be used for the ctkip activation code. If **DeviceserialIsCtkipCode** is set to **true**, the contents of the **DeviceSerial** field will be used for the ctkip activation code. Both options set to true will throw an error.

There are five optional parameters available to configure Soft Token deployments:

- Set **RegenerateSeed** to true to force the generation of new tokencodes or false to leave the current tokencodes in place. Default value is true if missing or empty.
- Set **OTPLength** to force the tokencode length. Valid values are **6** and **8** to force tokencode length to 6 digits and 8 digits respectively. Leave this parameter empty to generate the default value for tokens being deployed.

- Set OTPInterval to force the tokencode display duration. Valid values are 30, 60 and empty. 30 and 60 set the display duration to 30 or 60 seconds respectively. Empty will generate the default value for tokens being deployed.
- Set OTPAlgorithm to “Time” to configure the token as a time based token or to “Event” to configure the token as an event based token. Setting OTPAlgorithm to empty will generate the default value for tokens being deployed.
- Set PinAdded to “Passcode” to configure a soft token PIN as a passcode (pin-pad) token. In this mode, the PIN is added to the tokencode without carry. Set PinAdded to “tokencode” to configure a soft token PIN as a tokencode. In this mode, the PIN must be pre-appended to the token. Leave this parameter empty to generate the default value for tokens being deployed.

Automatic Notification

The following fields are used for automatic notification. When provisioning software tokens Automatic Notification may be used for delivery of the following information:

- For CT-KIP provisioning the activation code and delivery URL.
- For SDTID file provisioning the token sdtid file. Applies to both password protected files and unprotected files.
- For SDTID files that are password protected. The token file will be sent in one email and the password will be sent in a separate email.

For more information, see Appendix A, Automatic Notification.

DeliveryMethod - For email delivery set this field to one of the following values. Leave it empty to disable notification.

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

DestinationAddress - For SMTP this field can be used to provide an email address. If this field is empty or not used the principal’s account email address will be used.

TemplateFile - Use this field to provide the path and file name of an email template file. Template files are included in the AMBA package that may be used as a starting point. Providing a customized template allows additional text to be included in the email along with changes in format. If a custom template file is not provided, AMBA will use a built-in template.

InstanceName - Name of current instance. This is only required if **DeliveryMethod** is set to SMTP and multiple Instances have been declared for this server.

Software Token Device Type Attributes

A specific Software Token Device Type may be linked to a software token through the security console, the API or the AMBA SSTDT command. Software Token Device Types contain various sets of attributes which are basically name/value pairs. Default values may be assigned to these attributes through the security console, the APIs or AMBA. Use the variables declared in the following table to assign values to the attributes for specific tokens.

Software Token Device Type Attributes	AMBA Variable 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	SiteRUL2	0, value, empty
TOOLBAR_SITEURL3	SiteURL3	0, value, empty

Attribute values:

-2 - copy the TokSerial to the attribute value field.

-1 - copy the DefLogin to the attribute value field

0 - force 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

Note: Note: “User Password” is required to add user to IMS. Either each user can be provided with different password using “UserPwd” optional field or the cmd line option "--userpwd" can be used to have the same password given for all the users. If Identity Source is Internal Database and if no password is provided in both input file and as a command line option, Users will be added with password as null. If Identity Source is an External Directory like an LDAP, command will throw an Exception

Essentially, this command is identical to the Add User and Token command except you will supply a token type in place of the token serial. AMBA will then attempt to find an unassigned token of the requested type. If successful, the newly acquired token serial will be inserted in the Token Serial field and Add User and Token command will be called. If successful, the newly assigned Token Serial will be reported in the AMBA transaction log.

Add User and Password

The user will be added and a static password tokencode will be assigned to the user. The SetPin field specifies the initial password. If the user has been added previously a FAILURE message will be generated. The user may also be assigned to an existing Group

If the AddUser and Password function is successful, the password is automatically put into “new PIN mode” and the user must change the password at the first login.

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

If Security Domain and Identity Source are not provided, then the user will be added in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation. Group identified by “GrpName” need not be present under the same domain as the user, but should be under the same realm. A user may only be added to a group in the same IdentitySource or the Internal Database. If ForceGroupSearch is set “true”, a groupName will be searched for in the Internal Database. Any other value (including empty or missing) will cause the user’s IdentitySource to be searched for the group.

SetPin - User’s initial password — it must conform to system-defined PIN standards for number of characters and whether the characters are numeric or alphanumeric. The SetPin value becomes the user’s static password tokencode (a password used as a tokencode). It should not be confused with the IMS password defined in the following note.

Note: “User Password” is required to add a user to 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 cmd line option "--userpwd" can be used to have the same password given 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 will be added with an IMS password as null. If the Identity Source is an External Directory like an LDAP, the command will throw an Exception. Once 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 passcode if one is declared.

Add User Remote

Add user remote adds a remote user to the database.

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

If Security Domain is not provided, then the remote user will be added in the default security domain (System Domain) created during Authentication Manager installation. Trusted Group identified by “GrpName” need not be present under the same domain as the user, but should be under the same realm. A user may only be added to a group in the same IdentitySource or the Internal Database. If ForceGroupSearch is set “true”, a groupName will be searched for in the Internal Database. Any other value (including empty or missing) will cause the group to be searched for in the user's Identity Source.

IdentitySource needs to be provided if the SecurityDomain given in the input file is mapped to a different realm other than the default created during installation.

Add Token to User

The token to be added is specified in the ReplTokSerial field. The DefLogin field or the TokSerial field containing a token serial of a token already assigned to the user is used to identify the user. If both the fields are given then default login will be given precedence over TokSerial field. The specified Token will be assigned to the user.

The TokEnabled field defines the Token state following successful assignment. If the Token is already assigned or the user already has 3 Tokens assigned a FAILURE message will be generated.

PinMode may contain a value of 1 to set the token in new PIN mode or a value of 0 or empty to leave the new PIN mode as is. For more information on the SetPin and PinMode options, see Change PIN Status on page 67.

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

If “DefLogin” is given as input, and if Security Domain and Identity Source are not provided, then the user will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation. Token identified by the “ReplTokSerial” need not be present under the same domain as the user, but should be present in same realm as user.

If “TokSerial” is given as input, Identity Source and SecurityDomain values are not required and are ignored, if present. Consult the **--ctkip** command line option and the **SSTDT** command for options on generating ctkip credentials. For more information, see Set Soft Token Device Type on page 71.

If the --ctkip command line option has been declared and NicknameIsCtkipCode is set to “true”, the contents of the Nickname field will be used for the ctkip activation code. If DeviceserialIsCtkipCode is set to “true”, the contents of the DeviceSerial field will be used for the ctkip activation code. Both options set to true will throw an error.

There are five optional parameters available to configure Soft Token deployments:

- Set RegenerateSeed to true to force the generation of new tokencodes or false to leave the current tokencodes in place. Default value is true if missing or empty.
- Set OTPLength to force the tokencode length. Valid values are 6 and 8 to force tokencode length to 6 digits and 8 digits respectively. Leave this parameter empty to generate the default value for tokens being deployed.
- Set OTPInterval to force the tokencode display duration. Valid values are 30, 60 and empty. 30 and 60 set the display duration to 30 or 60 seconds respectively. Empty will generate the default value for tokens being deployed.
- Set OTPAlgorithm to “Time” to configure the token as a time based token or to “Event” to configure the token as an event based token. Setting OTPAlgorithm to empty will generate the default value for tokens being deployed.
- Set PinAdded to “Passcode” to configure a soft token PIN as a passcode (pin-pad) token. In this mode, the PIN is added to the tokencode without carry. Set PinAdded to “tokencode” to configure a soft token PIN as a tokencode. In this mode, the PIN must be pre-appended to the token. Leave this parameter empty to generate the default value for tokens being deployed.

Automatic Notification

The following fields are used for automatic notification. When provisioning software tokens Automatic Notification may be used for delivery of the following information:

- For CT-KIP provisioning the activation code and delivery URL.
- For SDTID file provisioning the token sdtid file. Applies to both password protected files and unprotected files.
- For SDTID files that are password protected. The token file will be sent in one email and the password will be sent in a separate email.

For more information, see Appendix A, Automatic Notification.

DeliveryMethod - For email delivery set this field to one of the following values. Leave it empty to disable notification.

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

DestinationAddress - For SMTP this field can be used to provide an email address. If this field is empty or not used the principal's account email address will be used.

TemplateFile - Use this field to provide the path and file name of an email template file. Template files are included in the AMBA package that may be used as a starting point. Providing a customized template allows additional text to be included in the email along with changes in format. If a custom template file is not provided, AMBA will use a built-in template.

InstanceName - Name of current instance. This is only required if **DeliveryMethod** is set to SMTP and multiple Instances have been declared for this server.

Software Token Device Type Attributes

A specific Software Token Device Type may be linked to a software token through the security console, the API or the AMBA SSTDT command. Software Token Device Types contain various sets of attributes which are basically name/value pairs. Default values may be assigned to these attributes through the security console, the APIs or AMBA. Use the variables declared in the following table to assign values to the attributes for specific tokens.

Software Token Device Type Attributes	AMBA Variable 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	SiteRUL2	0, value, empty
TOOLBAR_SITEURL3	SiteURL3	0, value, empty

Attribute values:

-2 - copy the TokSerial to the attribute value field.

-1 - copy the DefLogin to the attribute value field

0	- force 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

Add Token to User Automatic

This command automates the Add Token to User command by obtaining an unassigned token of a specified type from the system and calling the Add Token To User command using the newly acquired token serial number. The field definitions and requirements are identical to those of the ATU command with the following two exceptions:

The MiscVariable field is required and is used to supply the desired token type. Acceptable values are:

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

The ReplTokSerial field is not required and is ignored if present.

Only Non Expired Tokens will be searched and assigned to users. The MinTokenLife field is optional and is used to guarantee that a token will have a minimum number of days before its expiry date. The search for a token looks for tokens with an expiry date, greater than the current date. Adding a number of days with the MinTokenLife variable will adjust the search to tokens with an expiry date greater than today + MinTokenLife. For example, if MinTokenLife is set to 90, the token search would only get tokens with and expiry date greater than today + 90 days, meaning the token would not expire for at least 3 months or more. MinTokenLife is ignored if MiscVariable is set to -1.

Action	**ATUA**
Required Fields	DefLogin or TokSerial, TokEnabled, MiscVariable

Optional Fields	SetPin, PinMode, PinType, SoftIDParams, SoftIDPW, IdentitySource, SecurityDomain, Nickname, Filename, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, NicknameIsCtkipCode, DeviceserialIsCtkipCode, DeliveryMethod, DestinationAddress, TemplateFile, InstanceName

If “DefLogin” is given as input, if Security Domain and Identity Source are not provided, then the user will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation.

If “TokSerial” is given as input, Identity Source and SecurityDomain values are not required and are ignored, if present.

Token to be assigned will be searched in same realm as that of user while assigning token of a particular type. However if first available unassigned token is requested (-1) then the Authentication Manager API gets the token which expires first, irrespective of Security Domain.

If the --ctkip command line option has been declared and NicknameIsCtkipCode is set to “true”, the contents of the Nickname field will be used for the ctkip activation code. If DeviceserialIsCtkipCode is set to “true”, the contents of the DeviceSerial field will be used for the ctkip activation code. Both options set to true will throw an error.

There are five optional parameters available to configure Soft Token deployments:

- Set RegenerateSeed to true to force the generation of new tokencodes or false to leave the current tokencodes in place. Default value is true if missing or empty.
- Set OTPLength to force the tokencode length. Valid values are 6 and 8 to force tokencode length to 6 digits and 8 digits respectively. Leave this parameter empty to generate the default value for tokens being deployed.
- Set OTPInterval to force the tokencode display duration. Valid values are 30, 60 and empty. 30 and 60 set the display duration to 30 or 60 seconds respectively. Empty will generate the default value for tokens being deployed.
- Set OTPAlgorithm to “Time” to configure the token as a time based token or to “Event” to configure the token as an event based token. Setting OTPAlgorithm to empty will generate the default value for tokens being deployed.
- Set PinAdded to “Passcode” to configure a soft token PIN as a passcode (pin-pad) token. In this mode, the PIN is added to the tokencode without carry. Set PinAdded to “tokencode” to configure a soft token PIN as a tokencode. In this mode, the PIN must be pre-appended to the token. Leave this parameter empty to generate the default value for tokens being deployed.

Automatic Notification

The following fields are used for automatic notification. When provisioning software tokens, Automatic Notification may be used for delivery of the following information:

- For CT-KIP provisioning, the activation code and delivery URL.
- For SDTID file provisioning, the token .sdtid file. Applies to both password-protected files and unprotected files.
- For SDTID files that are password protected, the token file will be sent in one email and the password will be sent in a separate email.

For more information, see Appendix A, Automatic Notification.

DeliveryMethod - For email delivery set this field to one of the following values. Leave it empty to disable notification.

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

DestinationAddress - For SMTP this field can be used to provide an email address. If this field is empty or not used the principal's account email address will be used.

TemplateFile - Use this field to provide the path and file name of an email template file. Template files are included in the AMBA package that may be used as a starting point. Providing a customized template allows additional text to be included in the email along with changes in format. If a custom template file is not provided, AMBA will use a built-in template.

InstanceName - Name of current instance. This is only required if DeliveryMethod is set to SMTP and multiple Instances have been declared for this server.

Software Token Device Type Attributes

A specific Software Token Device Type may be linked to a software token through the security console, the API or the AMBA SSTDT command. Software Token Device Types contain various sets of attributes which are basically name/value pairs. Default values may be assigned to these attributes through the security console, the APIs or AMBA. Use the variables declared in the following table to assign values to the attributes for specific tokens.

Software Token Device Type Attributes	AMBA Variable Name	Permitted Values
Nickname	Nickname	-2, -1, 0, value, empty
DeviceSerialNumber	DeviceSerialNumber	-2, -1, 0, value, empty

Software Token Device Type Attributes	AMBA Variable Name	Permitted Values
TOOLBAR_SITEFILE_URL	SiteFile	0, value, empty
TOOLBAR_SITEURL1	SiteURL1	0, value, empty
TOOLBAR_SITEURL2	SiteRUL2	0, value, empty
TOOLBAR_SITEURL3	SiteURL3	0, value, empty

Attribute values:

-2 - copy the TokSerial to the attribute value field.

-1 - copy the DefLogin to the attribute value field

0 - force 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

Essentially, this command is identical to the Add Token To User command except you will supply a token type in place of the replacement token serial. AMBA will then attempt to find an unassigned token of the requested type. If successful, the newly acquired token serial will be inserted in the ReplTokSerial field and the Add Token To User command will be called. If successful, the newly assigned Token Serial will be reported in the AMBA transaction log. Consult the --ctkip command line option and the SSTDT command for options on generating CT-KIP credentials.

Add User to Group

The user associated with the specified Token or DefLogin will be added to the Group. Both the user and the group must exist. If the user is to have a different Login name or Shell within this Group it must be supplied via the optional fields, otherwise the User defaults will be used.

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

If Security Domain and Identity Source are not provided, then the user will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation. Group identified by “GrpName” need not be present under the same domain as the user, but should be under the same realm. For information on how to do a group search, see Add User on page 35.

Add Group

The Group specified in the GrpName field will be added.

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

If Security Domain and Identity Source are not provided, then the group will be added to the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation.

Add Group to Client

Enables a group of users on a client so that all members of the group can authenticate on that client. The function call must specify an existing group and client.

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

If Security Domain and Identity Source are not provided, then the group will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation.

Assign Profile

Assigns a profile specified by ProfileName to a user specified by TokSerial or DefLogin.

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

If “DefLogin” is given as input, and if Security Domain and Identity Source are not provided, then the user will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation. If “TokSerial” is given as input, then Identity Source and SecurityDomain values are not required and are ignored, if present.

Add Security Domain

Add a new Security Domain to the system.

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

PolicyType and PolicyName are numbered pairs. They may be used to select a specific policy in place of the system default. The number pairs may be declared in any order, however for every PolicyType there must be a matching PolicyName.

The valid PolicyTypes are:

PasswordPolicy

LockoutPolicy

SelfServicePolicy

AM_Token_Policy

AM_OFFLINE_AUTHN_POLICY(Case is significant)

Add Agent Host

This command is used to perform add, update, remove, list operations associated with the Agent Host.

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

The required Operation field allows one of the following functions to be performed:

OperationDetails

Add The AgentHost specified in the AgentHostName and/or AgentHostAdress field will be added.

Remove The operation will be used to remove the AgentHost, specified in the AgentHostName attribute.

List Produces a list of AgentHost information for each AgentHost. The AgentHost selection can be filtered with three input parameters AgentHostType, EnableFlag, AgentRestriction (if not specified all AgentHosts will be listed). The requested information will be written to the ResultsFile.

Update The operation will be used to update the attributes associated with the AgentHost, specified in the AgentHostName attribute.

Optional Fields

AgentHostName The attribute can be used with the Add/Remove/Update operations; it is mandatory for the Remove and Update operations.

For the Add operation, if either AgentHostName or AgentHostAddress is used, then an address or name lookup is attempted for the other. If both, fields are present, then no lookups are performed, and the values will be forced.

AgentHostAddress The attribute is used for the Add operation. If either AgentHostName or AgentHostAddress is used, then an address or name lookup is attempted for the other. If both fields are present, then no lookups are performed, and the values will be forced.

AgentHostType The attribute can be used for the Add and List operation, for adding a new AgentHost and filtering the results based on the AgentHostType.

The AgentHostType should be entered as a number between 1 and 4.

1 - Standard Agent

2 - Web Agent

3 - Radius Server Agent

4 - Radius Client Agent

If not specified for Add operation, it defaults to 1.

If not specified for List operation, all AgentHostType will be retrieved

AgentRestriction The attribute can be used for the Add/List/Update operation.

User Group Access Restriction.

0 - Restriction will be disabled

1 - Restriction will be enabled

If not specified for Add operation, it defaults to 0.

EnableFlag The attribute can be used for the Add/List/Update operation. Allowed values are true or false.

True - Agent is enabled.

False - Agent is disabled

If not specified for Add operation, it defaults to true.

Security Domain If SecurityDomain is not provided the default SecurityDomain (SystemDomain) will be used

ClearNodeSecret (***even if there is no NodeSecret***) The attribute can be used with the update operation, if set to true it can clear the node secret associated with the AgentHost. Allowed values are true or false.

True - clears the Node Secret

False - does not clear the Node Secret.

Not specified - defaults to False.

Single Softtoken Deployment

Action	SSD
Required Fields	TokSerial, TokEnabled
Optional Fields	Filename, SetPin, PinMode, PinType, softIDParams, SoftIDPW, IdentitySource, SecurityDomain, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, RegenerateSeed, OTPLength, OTPAlgorithm, OTPInterval, PinAdded, NicknameIsCtkipCode, DeviceserialIsCtkipCode, DeliveryMethod, DestinationAddress, TemplateFile, InstanceName

Deploys a single software token either in sdtid file format or through CTKIP. Prior to calling this function, the software token must be assigned to a user and the user must be in the RSA Authentication Manager internal database. Although the MSD command can be used to deploy a single token, it always creates a file titled <user ID>_<token serial>.sdtid wrapped in a zip file. The SSD command will create a <user ID>_<token serial>.sdtid only (not enclosed in a zip file). Additionally, the Filename parameter may be used to rename the file.

TokEnabled - This parameter is used to control the enabled/disabled state of the assigned token.

0 - Disable the assigned token (default)

1 - Enable the assigned token

SetPin - This parameter is used to initialize the PIN of the newly assigned token.

0 - Clear PIN. (Default) This action will automatically put the token in new PIN mode.

? - Any other value will be used as the new PIN.

Note: Assigning a literal value to **SetPin** will set all newly assigned tokens to this PIN.

PinMode - This parameter is used to control the new PIN mode of the newly assigned token.

0 - No action is taken (Default)

1 - Set the newly assigned token to new PIN mode.

PinType - This parameter is used to specify whether or not a token requires a pin.

Passcode - The token is a passcode token (requires a PIN)

Tokencode - The token is a tokencode only token (no PIN required)

SoftIDParams - When used, this field must contain three decimal digits (to maintain consistency with ACEBulkAdmin utility) that control the following seed file generation characteristics:

First digit:

0 - required but ignored

Copy Protection Flag 0 or 1:

0-Copy protection off

1 -Copy protection on

Password usage and Interpretation method 0, 1, 2, or 3:

0- No password

1- Static password (See SoftIDPW below)

2- Default login

3- Default login appended to static password

SoftIDPW- When used, this field supplies a password for the seed file encryption when a softID token seed record field is being created and **SoftIDParams** specifies a static password.

IdentitySource - It is only used to validate the security domain provided. If not provided, then the identity source mapped to the default security domain, for example, **Internal DataBase** will be used.

RegenerateSeed - For Soft Token deployments set this to true to force the generation of new tokencodes or false to leave the current tokencodes in place. Default value is true if missing or empty.

OTPLength - For Soft Token deployments use this parameter to force the tokencode length. Valid values are 6 and 8 to force tokencode length of 6 digits and 8 digits respectively. Leave this parameter empty to generate the default value for tokens being deployed.

SecurityDomain - The token will be searched for in the given security domain. If not provided, then the tokens will be searched for in the default security domain, for example, the System Domain created during installation.

NicknameIsCtkipCode & DeviceserialIsCtkipCode - If the --ctkip command line option has been declared and NicknameIsCtkipCode is set to “true”, the contents of the Nickname field will be used for the ctkip activation code. If DeviceserialIsCtkipCode is set to “true”, the contents of the Deviceserial field will be used for the ctkip activation code. Both options set to true will throw an error.

Automatic Notification

The following fields are used for automatic notification. When provisioning software tokens, Automatic Notification may be used for delivery of the following information:

- For CT-KIP provisioning the activation code and delivery URL.
- For SDTID file provisioning the token sdtid file. Applies to both password protected files and unprotected files.
- For SDTID files that are password protected. The token file will be sent in one email and the password will be sent in a separate email.

For more information, see Appendix A, Automatic Notification.

DeliveryMethod - For email delivery set this field to one of the following values. Leave it empty to disable notification.

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

DestinationAddress - For SMTP this field can be used to provide an email address. If this field is empty or not used the principal's account email address will be used.

TemplateFile - Use this field to provide the path and file name of an email template file. Template files are included in the AMBA package that may be used as a starting point. Providing a customized template allows additional text to be included in the email along with changes in format. If a custom template file is not provided, AMBA will use a built-in template.

InstanceName - Name of current instance. This is only required if DeliveryMethod is set to SMTP and multiple Instances have been declared for this server.

Note: The -g command line option is required to instruct AMBA to generate software token database files. Optionally, the -gdir command line option may be used to place any generated files in the specified directory. The --ctkip and -g options are mutually exclusive

The **--ctkip** command line option is required to instruct AMBA to generate CT-KIP credentials for qualifying tokens. The --ctkip and -g options are mutually exclusive. Also, the set soft token device type command (sstdt) can be used to force a specific device type for credential generation.

If the **-g** and --ctkip options are not used, the SSD command will not build any output files or generate any CT-KIP credentials. Additionally, if the **-g** command line option is not used, any **-gdir** command line option will be ignored.

Software Token Device Type Attributes

A specific Software Token Device Type may be linked to a software token through the security console, the API or the AMBA SSTDT command. Software Token Device Types contain various sets of attributes which are basically name/value pairs. Default values may be assigned to these attributes through the security console, the APIs or AMBA. Use the variables declared in the following table to assign values to the attributes for specific tokens.

Software Token Device Type Attributes	AMBA Variable 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	SiteRUL2	0, value, empty
TOOLBAR_SITEURL3	SiteURL3	0, value, empty

Attribute values:

-2- copy the TokSerial to the attribute value field.

-1- copy the DefLogin to the attribute value field

0- force 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 Change Functions

Change or Add User

This command is used to change the specified fields, however if the DefLogin is not present in AM, a new account will be created with a call to the Add User command. If the account does exist, the user data fields (LastName, FirstName and DefShell) will be modified if they are different than those in the database. If you wish to delete the FirstName or DefShell, supply a set of empty double quotes (“”) in the input field. LastName may not be deleted. If a GrpDefLogin is supplied, the user will be deleted from the group and added back in. This will result in the user being a member of the group with the supplied GrpDefShell (or its default). However, if the user was a member of the group under the DefLogin, this group membership will still exist. Empty double quotes are not applicable to the group.

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

If Security Domain and Identity Source are not provided, then the user will be searched/added in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation. A user may only be added to a group in the same IdentitySource or the Internal Database. If ForceGroupSearch is set “true”, a groupName will be searched for in the Internal Database. Any other value (including empty or missing) will cause the user’s IdentitySource to be searched for the group.

Note: “User Password” is required to add user to IMS. Either each user can be provided with different password using “UserPwd” optional field or the cmd line option "--userpwd" can be used to have the same password given for all the users. If Identity Source is Internal Database and if no is provided in both input file and as a command line option, Users will be added with password as null. If Identity Source is an External Directory like an LDAP, command will throw an Exception

For an existing user, leaving the UserPwd field blank will have no effect on the user’s IMS password. However if you wish to change the IMS password for a particular user, provide a value in UserPwd field. Check the command line option section and field definition section for more details.

Note: Although DefLogin is the only required field, at least one of the optional fields is required for anything meaningful to happen. Also, if the DefLogin does not exist in the database, the AU command is called, in which case the LastName field is also required.

Change or Add User and Token

This command has been added as a convenience to some users. It is recommended that extreme caution be used in selecting this command, as unintentional results are possible. This command is used to change the specified fields, however if the TokSerial is not assigned to a user account in the Authentication Manager Database, a new account will be created with a call to the Add User and Token command. If the TokSerial is associated with a user account, the user data fields (DefLogin, LastName, FirstName and DefShell) will be modified if they are different then those in the database. If you wish to delete a FirstName or DefShell, supply a set of empty double quotes (“”) in the input field. DefLogin and LastName may not be deleted. TokEnabled, SetPin fields will be updated if they are present and different from that found in the token record. The SetPin field is used to set the PIN state. If a GrpDefLogin is supplied along with GrpName, the user will be deleted from the group and added back in. This will result in the user being a member of the group with the supplied GrpDefShell (or its default). However, if the user was a member of the group under the DefLogin (or any other login), this group membership will still exist. Empty double quotes are not applicable to the group. For additional information on field values, see Field Definitions on page 23.

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

If Security Domain and Identity Source are not provided, then the user will be searched/added in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation. A user may only be added to a group in the same IdentitySource or the Internal Database. If ForceGroupSearch is set “true”, a groupName will be searched for in the Internal Database. Any other value (including empty or missing) will cause the user’s IdentitySource to be searched for the group.

If the --ctkip command line option has been declared and NicknameIsCtkipCode is set to “true”, the contents of the Nickname field will be used for the ctkip activation code. If DeviceserialIsCtkipCode is set to “true”, the contents of the Deviceserial field will be used for the ctkip activation code. Both options set to true will throw an error.

Automatic Notification

The following fields are used for automatic notification. Currently this service is only for delivery of CT-KIP credentials, via email.

DeliveryMethod - For email delivery set this field to one of the following values. Leave it empty to disable notification.

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

DestinationAddress - For SMTP this field can be used to provide an email address. If this field is empty or not used the principal’s account email address will be used.

TemplateFile - Use this field to provide the path and file name of an email template file. Template files are included in the AMBA package that may be used as a starting point. Providing a customized template allows additional text to be included in the email along with changes in format. If a custom template file is not provided, AMBA will use a built-in template.

Software Token Device Type Attributes

A specific Software Token Device Type may be linked to a software token through the security console, the API or the AMBA SSTDT command. Software Token Device Types contain various sets of attributes which are basically name/value pairs. Default values may be assigned to these attributes through the security console, the APIs or AMBA. Use the variables declared in the following table to assign values to the attributes for specific tokens.

Software Token Device Type Attributes	AMBA Variable 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	SiteRUL2	0, value, empty
TOOLBAR_SITEURL3	SiteURL3	0, value, empty

Attribute values:

-2	- copy the TokSerial to the attribute value field.
-1	- copy the DefLogin to the attribute value field
0	- force 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

Note: "User Password" is required to add user to IMS. Either each user can be provided with different password using "UserPwd" optional field or the cmd line option "--userpwd" can be used to have the same password given for all the users. Check the "CAU" command for more details. If Identity Source is Internal Database and if no password is provided in both input file and as a command line option, Users will be added with password as null. If Identity Source is an External Directory like an LDAP, command will throw an Exception

Note: Although TokSerial is the only required field, at least one of the optional fields is required for anything meaningful to happen. Also, if the TokSerial is unassigned, the AUT command is called, in which case, additional fields are required for a successful addition to occur. For more information, see Add User and Token on page 37. The PinMode field only affects the SetPin for changes as the API command called by AUT always forces new PIN mode.

Change or Add User and Password

This command is used to change the specified fields, however if the DefLogin is not present in the Authentication Manager Database, a new account will be created with a call to the Add User Password command. If the account does exist, the user data fields (LastName, FirstName and DefShell) will be modified if they are different then those in the database. If you wish to delete the FirstName or DefShell, supply a set of empty double quotes ("") in the input field. LastName may not be deleted. If a GrpDefLogin is supplied along with GrpName, the user will be deleted from the group and added back in. This will result in the user being a member of the group with the supplied GrpDefShell (or its default). However, if the user was a member of the group under the DefLogin (or any other login), this group membership will still exist. Empty double quotes are not applicable to the group field.

For additional information on field values, Field Definitions on page 23.

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

If Security Domain and Identity Source are not provided, then the user will be searched for or added in the default identity source (Internal DataBase) and security domain (System Domain) created, during Authentication Manager installation. A user may only be added to a group in the same IdentitySource or the Internal Database. If ForceGroupSearch is set “true”, a groupName will be searched for in the Internal Database. Any other value (including empty or missing) will cause the group to be searched for in the user's Identity Source.

SetPin - User’s initial IMS password — it must conform to system-defined PIN standards for number of characters and whether the characters are numeric or alphanumeric for the appropriate Security Domain. 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” (no quotes). For consistency, the aup command will not allow “disable” as a static password.

To modify a token PIN, see Change PIN Status on page 67.

Note: “User Password” is required to add a user to 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 cmd line option "--userpwd" can be used to have the same password given 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 will be added with an IMS password as null. If the Identity Source is an External Directory like an LDAP, the command will throw an Exception. Once 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.

Note: Although DefLogin is the only required field, at least one of the optional fields is required for anything meaningful to happen. Also, if the DefLogin does not exist in the database, the AUP command is called, in which case the additional fields are required for a successful addition to occur.

Change Principal Attribute Data

This command is used to modify **EXISTING** custom attribute data for custom attributes, assigned to a principal. The attributes can be either single-valued or multi-valued and can be of type Date, Integer, Float, Boolean, or String. Boolean attributes can only be single-valued, whereas Date, Integer, Float, and String types can be either single or multi valued.

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

If SecurityDomain and IdentitySource are not provided, the principal will be searched for in the default SecurityDomain (SystemDomain) and IdentitySource (InternalDatabase).

DefLogin identifies the principal (user) owning the attribute to be modified.

AttributeName identifies the custom attribute to be modified.

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

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

SubCommand is used to identify the specific operation to be performed on the attribute identified by AttributeName.

SubCommand values:

Single-valued attributes

S | s | empty	Set attribute to AttributeValue

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, you must use the Initialize (I) command to place the first value in the attribute list.

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

Failed to find target attribute: <attributeName>

After a multi-valued attribute has been initialized any of the other SubCommands may be used. If an “erase” SubCommand is used, the attribute must then be initialized if it is to be used again.

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

Change User/Token Security Domain

This command can be used to move a user and/or tokens to a new security domain. The command can move all users or individual users by default login. Optionally, all tokens, assigned tokens or no tokens can be moved to the new security domain.

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

If Security Domain and Identity Source are not provided, then the default identity source (Internal DataBase) and the default security domain (System Domain) will be used.

DefLogin:

If DefLogin contains an entry, that specific user account will be moved. If DefLogin contains the value “<all>” (without the quotes) all user accounts will be moved to the DestinationSecurityDomain. Case is not significant for <all>

DestinationSecurityDomain:

This variable is used declare the security domain to move the user account to and it must already exist in the Authentication Manager database.

MiscVariable:

This option is used to control whether or not tokens will be moved.

The possible values for MiscVariable are:

If DefLogin contains a userID:

0 - Move tokens assigned to specific user

1 - Do not move any tokens assigned to the specific user

If DefLogin contains the value <all>

0 - Move all users and all assigned tokens

1 - Move all users and no tokens

2 - Move all users and all tokens

3 - Move all unassigned tokens

4 - Move all assigned tokens

5 - Move all tokens

DefLogin	MiscVariable	Users Moved	Tokens Moved
specific userID	0	specific user	assigned to user[1]
specific userID	1	specific user	none

DefLogin	MiscVariable	Users Moved	Tokens Moved
<all>	0	all	assigned to user[1]
<all>	1	all	none
<all>	2	all	all
<all>	3	none	unassigned
<all>	4	none	assigned[2]
<all>	5	none	all[2]

[1] Tokens assigned to the user being moved will be 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.

SubDomain:

This option is used to indicate if the security domain search should include sub domains.

The possible values for SubDomain are:

0 - Do not include sub domains (the default value)

1 - Include sub domains.

Limit:

This option is used to limit the number of users and/or tokens to be moved. Set limit to a positive integer to set a maximum value for the number of users that will be moved. This provides a way to break up moving a very large number of users into a more manageable task. A Limit of 0 or empty implies no limit or "move all users." In addition to limiting the number of users that would be moved to a new security domain, the same value will limit the number of unassigned tokens that will be moved if that option is selected. Although there is no logical connection between users and unassigned tokens, using a limit for the number of unassigned tokens allows some control over the size of the task.

The SubDomain and Limit fields only apply when DefLogin is set to <all>. Otherwise, they are ignored.

Change User Remote

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 he or she should be authenticated in the specified remote realm using the specified login name (remoteAlias).

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

Note: There is no contact with the RSA Authentication Manager instance in the remote realm. The only changes are made locally. The login name in the remote realm (remoteAlias) is not verified.

If Security Domain is not provided, then the user will be searched in the default security domain (System Domain), created during Authentication Manager installation.

IdentitySource need to be provided, if the SecurityDomain given in the input file is mapped to a different realm, other than the default created during installation.

Change PIN Status

Change PIN Status provides the following functionality:

Clear the PIN associated with the specified token serial. The token will automatically be placed in new PIN mode.

Set the PIN associated with the specified token serial to an explicit value.

Force the specified token serial into new PIN mode.

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

The following values are valid for SetPin:

C or c or 0 Clear Pin for the specified token serial. The specified token serial is automatically forced to new PIN mode. PinMode is ignored.

Empty Force specified token serial into new PIN mode. This option is valid only if PinMode equals 1.

“N” New PIN Mode, Old PIN Required.

All other values Set PIN to supplied value.

Set PinMode equal to 1 to force the specified token serial into new PIN mode.

PIN attributes (min/max length, alpha and/or numeric, etc) are controlled by administrative settings. If you explicitly set a PIN and receive the ‘Invalid PIN’ error message, it may be a result of violating one or more of these attributes. The error message will not inform you as to the specific violation. In this case it will be necessary to check with an Authentication Manager administrator to find out what attributes have been set in your system.

Change Token Status

The specified assigned Token will be enabled or disabled according to the value of the TokEnabled parameter. Set TokEnabled to 1 in order to enable a token. Set TokEnabled to 0 (zero) in order to disable a token.

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

Change Token Status eXtended

The DefLogin or TokSerial will be used to locate all tokens assigned to a specific user. All tokens assigned to the user will be enabled or disabled according to the value of the TokEnabled parameter. Set TokEnabled to 1 in order to enable a token. Set TokEnabled to 0 (zero) in order to disable a token. The MiscVariable field is used to specify the token type to enable/disable.

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

0 - All tokens and passwords (Default)

1 - Tokens only (no passwords)

2 - Passwords only (no tokens)

SetPin, IdentitySource, SecurityDomain

In order to enable a static password it is now required to supply them via SetPin field. If no value is provided in SetPin field, Static Password will not be enabled even though TokEnabled is set to 1.

If “**DefLogin**” is given as input, if Security Domain and Identity Source are not provided, then the user will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation.

If “**TokSerial**” is given as input, Identity Source and SecurityDomain values are not required and are ignored, if present.

Change Token (Immediate)

If the Token specified in the ReplTokSerial field is currently unassigned it will immediately replace the Token specified in the TokSerial field.

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

Change Token (on First Use of New Token)

If the Token specified in the ReplTokSerial field is currently unassigned it will be added as a replacement Token to the User associated with the Token specified in the TokSerial field. The first time a user uses the new Token; the old Token will be disabled. The user's PIN is preserved.

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

This command is being depreciated and will be removed in a future release. Use the **REPT** or **REPTA** command instead. For more information, see Replace Token on page 79 or Replace Token Automatic on page 82.

Change Temporary User

Puts a user (identified by token serial number or login) in temporary mode. You can set both starting and ending times, or you can set only an ending time (.temporary mode lasts from now until the ending time). Both date and hour must be specified for starting and ending times. Leaving all optional fields empty or specifying 00/00/0000 in the DefShell field will remove the user from temporary status.

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

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

If “DefLogin” is given as input, and if Security Domain and Identity Source are not provided, then the user will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation. If “TokSerial” is given as input, then Identity Source and SecurityDomain values are not required and are ignored, if present.

Set Emergency Access Fixed

Sets the status of token (identified by a token serial number) to Fixed and assigns a fixedpassword. The lifetime of the fixed password can be defined in local time by using either the dateExpire, hourExpire, or lifeTime arguments.

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

The desired password is supplied in the SetPin field. Its acceptability is governed by system parameters.

An expiration date may be supplied in the CreatePin field. If a date is given, then optionally an hour entry may be included. Additionally, an optional lifetime entry (hours) may be included in the CreatePin field. If a lifetime entry is included, any date entry will be ignored. The format for the CreatePin entry is Dmm/dd/yyyyHnLn. Any Date, Hours, and Lifetime entry must appear in that order, DHL. 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. The only reason HnLn is valid, is because when a Lifetime entry is present, the Date and Hour entries are ignored.

Set Emergency Access OTP

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

Action	**EAOTP**
Required Fields	TokSerial

Optional Fields	SetPin, CreatePin

SetPin may be used to specify number of passwords to generate. Format for setPin is

Nn where n represents the number of one time passwords to generate.(Max is 50).Length and Format of One time passwords depend on Token policy.

An expiration date may be supplied in the CreatePin field. If a date is given, then optionally an hour entry may be included. Additionally, an optional lifetime entry (hours) may be included in the CreatePin field. If a lifetime entry is included, any date entry will be ignored. The format for the CreatePin entry is Dmm/dd/yyyyHnLn. Any Date, Hours, and Lifetime entry must appear in that order, DHL. 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. The only reason HnLn is valid, is because when a Lifetime entry is present, the Date and Hour entries are ignored.

Set Emergency Access OFF

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.

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

Set Soft Token Device Type

Sets a Soft Token Device Type to be used for soft token attribute assignment and CT-KIP provisioning. Setting a Soft Token Device Type causes any device type associated with a soft token to be replaced with the specified device type. This may be used for CT-KIP credential generation and/or any attribute assignments. The new device type is in effect for the remainder of the process or until it is replaced with a new SSTDT command or cleared by a new SSTDT command.

Note: This command affects any soft token assignment during a single AMBA process. This may happen as a result of the following commands (aut, auta, atu, atua, msd, mta and mtr). During CT-KIP provisioning or token assignment, the token will have its soft token device type forced to the specified device type. This change of soft token device type is saved in the database. Once saved in the database, it need not be re-specified. Any SSTDT specification is not saved in the database and must be re-specified in successive AMBA runs.

Action	**SSTDT**
Required Fields	None
Optional Fields	Key, KeyType, MiscVariable

Key String

Specifies the Family (Name) or Label (description) of the device type. This value is used to search the database for the Soft Token Device Type. If this value is empty or the Key column is missing, the Soft Token Device Type is cleared thereby cancelling any further action. Additionally, if a reset or clearing operation has been specified, KeyType and MiscVariable will be ignored. ***This value is Case Sensitive!***

KeyType

“FamilyKey”, “LabelKey”, empty or null (missing). Indicates the search argument supplied in key. If empty or null (missing), “FamilyKey” is assumed.

MiscVariable

String, empty or null (missing). The version of the device type to search for. If empty or null (missing) “1.0” is assumed.

SSTDT must be used in conjunction with the CIF command to issue software tokens. The following is a sample of an AMBA script to assign software tokens via CT-KIP:

```
Action,Key,KeyType,MiscVariable
SSTDT,BlackBerry,FamilyKey,3.5
CIF,Action,IdentitySource,DefLogin,MiscVariable,TokEnabled,PinMode,DeliveryMethod,DeviceSerialNumber,Nickname,Filename,PinType,PinAdded,SoftIDParams,SoftIDPW,TemplateFile
ATUA,<IDSOURCE>,<USERID>,4,1,1,SMTP2,<BlackberryPIN>,<IDSOURCE>-<USERID>,<USERID>.sdtid,passcode,tokencode,1,<PASSWORD>,/opt/rsa/am/amba/SDTIDDefaultNotice.xml
```

Update User Data

This command 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 the old DefLogin in the MiscVariable field. TokSerial takes precedence over MiscVariable. If the TokSerial field is not present and MiscVariable contains a value, ABMA will attempt to obtain a TokSerial using this value. If a TokSerial is obtained, the DefLogin will be changed.

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

Note: If Security Domain and Identity Source are not provided **and if DefLogin is provided as required field**, then the user will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation.

However if TokSerial Field is provided in required field, then Security Domain and Identity Source are not required and are ignored if provided.

Note: This command can also be used to update IMS Password for a user. Either each user can be provided with different password using “UserPwd” optional field or the cmd 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

This command updates the Administrative Roles associated with the user. The command can be used to add/remove/clear Admin Roles associated with a user.

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

For assigning an Admin Role to the User, the valid Operation value along with a DefLogin and the RoleName needs to be provided.

For unassigning an Admin Role from a user, the valid Operation value along with a DefLogin and the RoleName needs to be provided.

To clear all the roles which are currently assigned to the user, the valid Operation value along with a DefLogin needs to be provided.

RoleName: Name of the Admin role which needs to searched.

RoleSecurityDomain: Security Domain in which the Role needs to be searched.

(Defaults to System Domain)

Operation:

Values	Details
Add	Assigns the Admin Role to the User.
Remove	Unassigns the Admin Role to the User.
Clear	Unassigns all Admin Roles for the User.

Note: Note: If Security Domain and Identity Source are not provided, then the user will be searched in the default identity source (Internal DataBase) and security domain (SystemDomain) created during Authentication Manager installation. Similarly, if the RoleSecurityDomain is not provided, the Role will be searched in the default security domain (System Domain) created during Authentication Manager installation.

Change Token Attributes

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

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

For assigning a Token Attribute to the Token, the valid Operation value along with a TokSerial and the AttributeName needs to be provided.

For unassigning a Token Attribute from a Token, the valid Operation value along with a TokSerial and the AttributeName needs to be provided.

To update the token attribute, the valid Operation value along with a TokSerial and the AttributeName needs to be provided. If a token attribute name needs to be updated, then a DestinationAttributeName needs to be provided, along with the new AttributeValue.

Operation Must be one of the following functions:

Add Assigns the token attribute with the Token

Update modifies the Token Attribute Name and Value, associated with the Token.

Clear Unassign the Token Attribute associated with the Token.

AttributeName: Custom Token Attribute Name to be searched (defined under Token Attribute Definitions in Authentication Manager).

AttributeValue: Attribute value to be associated with the Toke Attribute.

DestinationAttributeName: New Attribute Name

Note: If the AttributeValue is not passed, the default value associated with the Custom Token Attribute (declared while defining the Token Attribute Definition) will be populated. If both values (AttributeValue and DefaultValue) happen to be blank an error message will be thrown.

Change Token Security Domain

This command can be used to move specific token or range of tokens to a new security domain.

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

TokSerial

The variable is used to declare the specific Token Serial to be moved, or could be the starting range for the lists of tokens to be moved (if the EndRange variable is specified).

DestinationSecurityDomain

This variable is used to declare the security domain where the tokens will be moved to, and it must already exist in the Authentication Manager database.

EndRange

The variable is used to declare the ending Token Serial number in the range. All the tokens starting from the TokSerial variable and EndRange value will be moved.

SecurityDomain

The variable is used to declare the security domain where the tokens will be searched. If SecurityDomain is not provided the default security domain (SystemDomain) will be used.

TokenType

The variable is used to limit the token lookup, the below set of values are allowed. If not specified all type of tokens will be searched.

The possible values for TokenType are:

0 - RSA SecurID Standard Card

1 - RSA SecurID PINPAD Card

2 - RSA SecurID Key Fob

3 - RSA SecurID Watch

4 - RSA SecurID Software Token (formerly SoftID)

5 - RSA SecurID Smartcard

6 - RSA SecurID Modem

7 - RSA SecurID Crypto

8 - RSA SecurID Proteus

9 - RSA SecurID USBCOSMO (SID800)

10 - RSA SecurID Flextoken

If not specified all the tokens will be moved irrespective of the Type.

TokenAssigned

The variable is used to indicate whether the list of tokens to be moved, are either Assigned or Unassigned. By default all the unassigned tokens will be searched.

The possible values for TokenAssigned are:

0 - Unassigned (the default value)

1 - Assigned Tokens

SubDomain

This option is used to indicate if the security domain search should include sub domains.

The possible values for SubDomain are:

0 - Do not include sub domains (the default value)

1 - Include sub domains.

Limit

This option is used to limit the number of tokens to be moved. This provides a way to break up moving a very large number of tokens in the range specified into a more manageable task.

5 Delete Functions

Delete User from Group

The User associated with the DefLogin field will be removed from the Group.

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

If Security Domain and Identity Source are not provided, then the user will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation. A user may only be added to a group in the same IdentitySource or the Internal Database. If ForceGroupSearch is set “true”, a groupName will be searched for in the Internal Database. Any other value (including empty or missing) will cause the group to be searched for in the user's Identity Source.

Delete Group from Client

The Group specified will be removed from the Client.

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

If Security Domain and Identity Source are not provided, then the group will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation.

Delete Group

Deletes a specified group record from the database.

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

If Security Domain and Identity Source are not provided, then the group will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation.

Delete Security Domain

Delete Security Domain from the system.

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

Delete Token

Deletes from the database the record of an **unassigned** token identified by TokSerial.

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

Delete User

The specified user will be deleted from the database and any associated Tokens returned to the unassigned state. The user is also removed from all other associations (Groups etc).

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

If Security Domain and Identity Source are not provided, then the user will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation.

Unassign Profile

Unassign the currently assigned user profile.

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

If “DefLogin” is given as input, and if Security Domain and Identity Source are not provided, then the user will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation.

If “TokSerial” is given as input, then Identity Source and SecurityDomain values are not required and are ignored, if present.

Replace Token

The TokSerial field contains the token serial number of the token to be replaced and must already be assigned to a user. The ReplTokSerial field contains the token serial number of the token replacing TokSerial. The token specified by ReplTokSerial will be assigned to the user as a replacement token. The TokEnabled field defines the Token state following successful assignment. If the ReplTokSerial token is already assigned, a FAILURE message will be generated. PinMode may contain a value of 1 to set the token in new PIN mode or a value of 0 or empty to not change the new PIN mode.

Action	**REPT**
Required Fields	TokSerial, ReplTokSerial
Optional Fields	TokEnabled, SetPin, PinMode, PinType, SoftIDParams, SoftIDPW, Filename, copyProtect, IdentitySource, SecurityDomain, SiteFile, SiteURL1, SiteURL2, SiteURL3, DeviceSerialNumber, DeviceserialIsCtkipCode, Nickname, NicknameIsCtkipCode, RegenerateSeed, OTPLength, OTPAlgorithm, OTPInterval, PinAdded, DeliveryMethod, DestinationAddress, TemplateFile

The token identified by the “ReplTokSerial” need not be present under the same domain as the user, but should be present in the same realm as the user.

Consult the ctkip command line option and the sstdt command for ctkip credentials generation options.

Note: The -g command line option is required to instruct AMBA to generate software token database files. Optionally, the -gdir command line option may be used to place any generated files in the specified directory. The --ctkip and -g options are mutually exclusive

The **--ctkip** command line option is required to instruct AMBA to generate CT-KIP credentials for qualifying tokens. The --ctkip and -g options are mutually exclusive. Also, the set soft token device type command (sstdt) can be used to force a specific device type for credential generation.

If the -g and --ctkip options are not used, the REPT command will assign tokens but will not build any output files or generate any CT-KIP credentials. Additionally, if the -g command line option is not used, any -gdir command line option will be ignored.

IdentitySource - This argument is only used to validate the security domain provided. Users provided will be searched in the given identity source. If not provided, then the users will be searched in the identity source mapped to the default security domain, for example, the **Internal DataBase** created during installation.

SecurityDomain - Tokens will be searched in the given security domain. If not provided, then the tokens will be searched in the default security domain, for example, the **System Domain** created during installation.

TokSerial - Specifies the serial number of the token to be replaced.

ReplTokSerial - Specifies the serial number of the replacement token.

TokEnabled - This parameter is used to control the enabled/disabled state of the replacement token.

0 - Disable the assigned token (default)

1 - Enable the assigned token

PinMode - This parameter is used to control the new PIN mode of the newly assigned token. If PinMode is 0 or not set at all, new PIN mode will be reset (forced to off) for the newly assigned token. This will cause each newly assigned token to inherit the PIN and the new PIN mode state of the token it is replacing. If PinMode is set to 1, each newly assigned token will be placed in new PIN mode, its PIN will be cleared and will NOT inherit the PIN of the token being replaced.

0 - Replacement token will inherit the PIN and new PIN mode state of the token it is replacing.

1 - Clears the PIN of the newly assigned token and forces new PIN mode on.

PinType - This parameter is used to specify whether or not a token requires a pin.

Passcode - The token is a passcode token (requires a pin)

Tokencode - The token is a tokencode only token (no PIN required)

SoftIDParams - When used, this field must contain three decimal digits (to maintain consistency with ACEBulkAdmin utility). This parameter is ignored if **ctkip** is true. **SoftIDParams** controls the following seed file generation characteristics:

First digit:

0 - required but ignored

Copy Protection Flag 0 or 1:

0-Copy protection off

1 -Copy protection on

Password usage and Interpretation method 0, 1, 2, or 3:

0- No password

1- Static password (See SoftIDPW below)

2- Default login

3- Default login appended to static password

SoftIDPW - Usage depends on third digit of **SoftIDParams**. When used, this field supplies a password be provided by in order to access a RSA SecurID software token **sdtid** file. This field supplies a password to be used for the seed file encryption when **SoftIDParams** specifies a static password. If there is no password associated with the file, an empty string may be passed. This parameter is ignored if **ctkip** is true.

filename - Specifies the name of the output file. If left empty, the fileName defaults to “<user id>_<token serial>.sdtid. Ignored if --ctkip is true.

copyProtect - Specifies whether the copy protection is enabled for the software token. If copy protection is enabled, the Software Token record cannot be removed from the directory in which it is installed on a user’s computer. If 0, copy protection is disabled; if any other value, copy protection is enabled. Ignored if --ctkip is true.

Automatic Notification

The following fields are used for automatic notification. When provisioning software tokens Automatic Notification may be used for delivery of the following information:

For CT-KIP provisioning, the activation code and delivery URL.

For SDTID file provisioning, the token sdtid file. Applies to both password- protected files and unprotected files.

For SDTID files that are password-protected. The token file will be sent in one email, and the password will be sent in a separate email.

For more information, see Appendix A, Automatic Notification.

DeliveryMethod - For email delivery, set this field to one of the following values. Leave it empty to disable notification.

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

DestinationAddress - For SMTP this field can be used to provide an email address. If this field is empty or not used the principal's account email address will be used.

TemplateFile - Use this field to provide the path and file name of an email template file. Template files are included in the AMBA package that may be used as a starting point. Providing a customized template allows additional text to be included in the email along with changes in format. If a custom template file is not provided AMBA, will use a built-in template.

InstanceName - Name of current instance. This is only required if DeliveryMethod is set to SMTP and multiple Instances have been declared for this server.

Software Token Device Type Attributes

A specific Software Token Device Type may be linked to a software token through the security console, the API or the AMBA SSTDT command. Software Token Device Types contain various sets of attributes which are basically name/value pairs. Default values may be assigned to these attributes through the security console, the APIs or AMBA. Use the variables declared in the following table to assign values to the attributes for specific tokens.

Software Token Device Type Attributes	AMBA Variable 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	SiteRUL2	0, value, empty
TOOLBAR_SITEURL3	SiteURL3	0, value, empty

Attribute values:

-2 - copy the TokSerial to the attribute value field.

-1 - copy the DefLogin to the attribute value field

0 - force 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

Replace Token Automatic

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

Action	**REPTA**
Required Fields	TokSerial, MiscVariable
Optional Fields	TokEnabled, SetPin, PinMode, PinType, SoftIDParams, SoftIDPW, Filename, copyProtect, IdentitySource, SecurityDomain, SiteFile, SiteURL1, SiteURL2, SiteURL3, DeviceSerialNumber, DeviceserialIsCtkipCode, Nickname, NicknameIsCtkipCode, RegenerateSeed, OTPLength, OTPAlgorithm, OTPInterval, PinAdded, DeliveryMethod, DestinationAddress, TemplateFile

The field definitions and requirements are identical to those of the REPT command except for the following exceptions:

ReplTokSerialThe ReplTokSerial field is not required and is ignored, if present.

MiscVariable - The MiscVariable field is required and is used to supply the desired token type for the replacement token. Acceptable values are:

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

MinTokenLife - The MinTokenLife field is optional and is used to guarantee that a replacement token will have a minimum number of days before its expiry date. The search for a replacement token looks for tokens with a expiry date greater than the current date. Adding a number of days with the MinTokenLife variable will adjust the search to tokens with an expiry date greater than today + MinTokenLife. For example, if MinTokenLife is set to 90, the replacement token search would only get tokens with and expiry date greater than today + 90 days, meaning the token would not expire for at least 3 months or more. MinTokenLife is ignored if MiscVariable is set to -1.

IdentitySource - This argument is only used to validate the security domain provided. Users provided will be searched in the given identity source. If not provided, then the users will be searched in the identity source mapped to the default security domain, for example, the **Internal DataBase** created during installation.

SecurityDomain - Tokens will be searched in the given security domain. If not provided, then the tokens will be searched in the default security domain, for example, the **System Domain** created during installation.

Token to be assigned will be searched in same realm as that of user while assigning token of a particular type. However if first available unassigned token is requested (-1) then the Authentication Manager API gets the token which expires first, irrespective of Security Domain.

Essentially, this command is identical to the Replace Token command except you will supply a token type in place of the replacement token serial. AMBA will then attempt to find an unassigned token of the requested type. Additionally, you may specify a minTokenLife which will guarantee the replacement token has at least that many days of life before it is no longer usable. If successful, the newly acquired token serial will be inserted in the ReplTokSerial field and the Replace Token command will be called. If successful, the newly assigned Token Serial will be reported in the AMBA transaction log. Consult the ctkip command line option and the sstdt command for ctkip credentials generation options.

Rescind Token

The token specified will be unassigned. No other action regarding this user or token is performed.

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

Unassign Token

Unassigns a token from a 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. If you only want to unassign the token and leave the user account in place, use Rescind Token on page 84

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

6 On-Demand Authentication Functions

This section describes functions related to “OnDemand Authenticators” as opposed to “tokens.” The use of these commands and on-demand authentication in general requires proper configuration settings in RSA Authentication Manager to enable either SMS or SMTP (email) Tokencode Delivery. At least one delivery method must be configured. Setup is accomplished through the Security Console by clicking **Setup** > **System Settings** > **Authentication Settings** > **On-Demand Tokencode Delivery**

Configure either SMS, SMTP (Email) or both depending on your needs. Use the Tokencode settings tab to modify the message format or the default tokencode lifetime. If only one delivery method is configured, the enable command will automatically select that method. If both delivery methods are configured, a deliveryMethod parameter must be supplied with the enable command to indicate which delivery method is to be used.

Enable OnDemand Authentication

This command enables an OnDemand Authenticator for a principal. A PIN may be assigned or system generated. Some field names are duplicates of field names used for token management but may have slightly different usage. Consult this guide and the RSA Authentication Manager Help for the correct definition of ODA command fields. Results are written to the AMBA results file. Use the CRFN command to change the results file path and name.

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

* These fields are ignored when PINIndicator is set to GENERATE_PIN.

Field Definitions:

DefLogin	principal user ID	
PINIndicator	SET_TEMP_PIN	ODA is initialized with provided PIN. Authenticator is in ‘New Pin Mode.’
	SET_PERM_PIN	ODA is initialized with 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 principal’s token policy.
	NO_PIN_UPDATE	Option that indicates that the user's PIN is not being modified.

IdentitySource IdentitySource of principal. Defaults to InternalDatabase.

SecurityDomain SecurityDomain of principal. Defaults to SystemDomain.

InstanceName Name of current instance. This is only required if Notify is set as an outputOption and multiple Instances have been declared.

SetPin Initial PIN value when enabling and ODA PINIndicator set to SET_TEMP_PIN or SET_PERM_PIN.

PinMode True for newPin mode, otherwise false. Default is false.

ExpiryDate Date the OnDemand Authenticator will automatically expire. Format is yyyy.mm.dd.HH.mm z

DeliveryMethod SMS or SMTP

DestinationAddress The SMTP (email) address or the SMS number depending on the DeliveryMethod.

TemplateFile The XML template file to use for constructing the email. If empty, AMBA uses a default template. There are default template files included with the AMBA distribution that may be used as a starting point for building your own.

OutputOption Any combination of the following:

H | h Write a header line to the results file

A | a Append output to an existing results file.

O | o Omit PIN from results file. (replaced by <omitted>)

N | n Notify. Send SMTP or SMS message with generated pin to principal. Message type depends on deliveryMethod. (SMS currently not implemented.)

Update OnDemand Authentication

This command updates existing OnDemand Authentication settings of a principal. Some field names are duplicates of field names used for token management but may have slightly different usage. See the Security Console Help for the correct definition of ODA command fields. Results are written to the AMBA results file. Use the CRFN command to change the results file path and name.

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

* PINIndicator does not support GENERATE_PIN for UODA.

** These fields are ignored when PINIndicator is set to NO_PIN_UPDATE.

Field Definitions:

DefLoginprincipal user ID

PINIndicatorSET_TEMP_PIN - ODA is updated with provided pin. Authenticator is in ‘New Pin Mode.’

SET_PERM_PIN - ODA is updated with provided pin. Authenticator is **not** put into ‘New Pin Mode.’

NO_PIN_UPDATE - Option that indicates that the user's PIN is not being modified.

IdentitySource - IdentitySource of principal. Defaults to InternalDatabase.

SecurityDomain - SecurityDomain of principal. Defaults to SystemDomain.

InstanceName - Name of current instance. This is only required if Notify is set as an outputOption and multiple Instances have been declared.

SetPin - Initial PIN value when enabling and ODA PINIndicator is

set to SET_TEMP_PIN or SET_PERM_PIN.

PinMode - True for newPin mode, otherwise false. Default is false.

ExpiryDate - Date the OnDemand Authenticator will automatically expire. Format is yyyy.mm.dd.HH.mm z

DeliveryMethod - SMS or SMTP

DestinationAddress - The SMTP (email) address or the SMS number depending on the deliveryMethod.

TemplateFile - The XML template file to use for constructing the email. If empty, AMBA will use a default template. The default template files included with the AMBA distribution may be used as a starting point for building your own.

OutputOption - Any combination of the following:

H | h	Write a header line to the results file
A | a	Append output to an existing results file.
O | o	Omit PIN from results file. (Replaced by <omitted>)
N | n	Notify. Send SMTP or SMS message with generated PIN to principal. Message type depends on deliveryMethod. (SMS currently not implemented.)

Disable OnDemand Authentication

This command disables an OnDemand Authenticator for a principal. Results are written to the AMBA results file. Use the CRFN command to change the results file path and name.

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

Field Definitions:

DefLogin	principal user ID
IdentitySource	IdentitySource of principal. Defaults to InternalDatabase.

SecurityDomain SecurityDomain of principal. Defaults to SystemDomain.

OutputOption Any combination of the following:

H | h Write a header line to the results file

A | a Append output to an existing results file.

7 List Functions

Overview

The list functions provide a powerful tool for producing data that can be used as is or passed on to some other product for additional sorting, formatting or combining with data from other sources. The output of the list functions 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 functions start out by extracting 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. Based on additional input parameters, additional data such as all user information, extended user information, user extension data and group membership information may be reported. Token information for assigned tokens may also be included. For token-based reports, full token information and token extension data may be reported along with user and group information if the token is assigned to a user.

The listTokenInfoByField command is particularly powerful in that in addition to token information it provides for option of appending user information for assigned tokens. For example, this command could be used to obtain a list of all tokens expiring in 45 days along with the user and user extension data for this token. The list could then be used to generating a mailing or to automatically email these users. Of course, this would depend on the necessary information being kept in user extension data fields.

Most of the list reporting functions are also available to the multiple token assignment and multiple token replacement commands. If these commands are used to assign or replace tokens, the report functions can be used for token control or notification.

The output of these commands is a comma separated variable list that can easily 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 will be the single word ‘Done’ followed by a newline sequence. In this case, a report that generated no output will consist of a single line beginning with the word ‘Done.’

List Secondary Nodes for Agent Host

This function will list all of the secondary nodes associated with a given agent host. The list is in comma separated variable (CSV) format and is written to the results file. (See the -r command line argument for additional information on the results file).

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

If Security Domain is not provided, then the Agent Host will be searched in the default security domain (System Domain) created during Authentication Manager installation.

IdentitySource need to be provided, if the SecurityDomain given in the input file is mapped to a different realm other than the default created during installation.

List User Info by Field

This command will produce a list of default logins or a list of user information for each default login. The default logins selected are based on three input parameters, CompareField, CompareType, and CompareValue. Additionally, the listing is controlled by three additional parameters. The requested information is written to the ResultsFile. See the CRFN command for information on changing the results file name.

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

If Security Domain and Identity Source are not provided, then the user will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation.

Select the CompareField and the CompareType values from the following. Supply a value for CompareValue of the type listed in the value column. The default values for CompareField, CompareType, and CompareValue are 0, 0, and 0. If either CompareField or CompareType is 0, the other is assumed to be 0.

User Listed	Field	Type	Value
All	0	0	Ignored
By last name	1		
All beginning with		1	String
All matching		2	String

User Listed	Field	Type	Value
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
Local or remote	5		
All local		1	Ignored
All remote*		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**
All with tokens expiring in # of days		7	Number

User Listed	Field	Type	Value
By LDAP data***	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

* For Remote Users standard header in 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 will be ignored.

** Token type values:

0	RSA SecurID Standard Card
1	RSA SecurID PINPAD Card
2	RSA SecurID Key Fob
3	RSA SecurID Watch
4	RSA SecurID Software Token (formerly SoftID)
5	RSA SecurID Smartcard
6	RSA SecurID Modem
7	RSA SecurID Crypto
8	RSA SecurID Proteus
9	RSA SecurID USBCOSMO (SID800)
10	RSA SecurID Flextoken

***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 AMBA aligned as closely as possible with the ACEBulkAdmin.

The **OutputOption** may contain the following values:

Default for all versions is 0. List user information for all selected default logins; do not write a header record.

[[-2 | -1 | 0 - 7 | 30 - 34] [H | h] [A | a]]

where:

-2   produces a listing of default logins (no data) and any assigned token serial numbers (no data).

-1   produces a listing of default logins only (no data)
0   list user information for selected default logins

1   append LDAPsource to option 0

2   append remote alias to option 0

3   append profile name to option 0

4   option 0, 1, 2, and 3 combined

5   LDAPsource only

6   remote alias only

7   profile name only

30   option 0 plus token info for assigned tokens

31   option 1 plus token info for assigned tokens

32   option 2 plus token info for assigned tokens

33   option 3 plus token info for assigned tokens

34   option 4 plus token info for assigned tokens

60   option 30 plus replacement info for assigned tokens*

H | hwrite a header record in the output file

A | aopen results in append mode.

*Option 60 is a special case. It is the same as option 30 except 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 may be used simultaneously, which may result in heading lines within the file body. This choice is intentional and it is up to the user to determine which options make sense and whether or not down stream applications will be able to correctly process this file. Also, because CIF commands may be used between list commands, if the append option is used, data columns may not line up. Again, this action is intentional and the user must choose the options appropriate for the desired outcome.

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

[0 | 1 | 2 . . . | 11]

Where:

no additional data listed

include token extension data

include user extension data

option 1 and 2

include group membership group names

option 1 and 4

option 2 and 4

option 3 and 4

include group membership field values

option 1 and 8

option 2 and 8

option 3 and 8

Because these options may 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 will be 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 & 6) or the first group membership default login (options 8 & 10) will be 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 will be appended after any of the above information. Each token will be listed as *tagBTOK*, token info. If token extension data has been requested, it will follow the applicable token as *tagBTED*, followed by as many token extension data key/data pairs as exist for this token.

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 may appear more than once. When appropriate, the special identifier tag names will also appear. Header records for the ListTokenInfoByField are as follow:

The following field identifiers will be included in the heading, if their associated data has been 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 have been requested, the data, is present, will appear in “pairs” for as many assigned tokens a user may have. For example, a user with two assigned tokens, each with extension data would have a listing appended to any of the above requested information something like the following:

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

The MiscVariable is used to fix up extension keys and/or data that contain commas. Some users have either intentionally or unintentionally entered commas in their extension data keys or data. Because the retrieval of the extension data and keys is returned as a comma separated list, what is the key and what is the data can be next to impossible to figure out. If you are requesting extension data in this list, AMBA attempts to fix up key/data pairs that contain more than one comma. It does so by parsing the results, building 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. Once a key if 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 will be valid as either the first portion only, or any concatenated portions. In any case, MiscVariable may be used to control the substitution. MiscVariable may 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

If the MiscVariable is either absent or empty, the default action is to replace extra commas by semi-colon.

List User Info for User

This command will produce user and assigned token information for one user. This command duplicates the List User Info by Field command (above) in every respect except one. The difference is that this command will supply the information for a single supplied default login or an associated user if a token serial number is supplied. If both are supplied, the token serial number is ignored.

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

If Security Domain and Identity Source are not provided **and if DefLogin is provided as required field**, then the user will be searched in the default identity source (Internal DataBase) and security domain (System Domain) created during Authentication Manager installation.

However if TokSerial Field is provided in required field, then Security Domain and Identity Source are not required and are ignored if provided.

Consult the List User Info by Field command (above) for the definition of the optional fields usage and output formatting options.

List Token Info by Field

This command will produce a list of token serial numbers or a list of token information for each token serial number. The token serial numbers selected are based on three input parameters, CompareField, CompareType, and CompareValue. Additionally, the listing is controlled by three additional parameters. The requested information is written to the ResultsFile. See the CRFN command for information on changing the results file name.

If a selected token is assigned to a user, user information and group information may also be included in the output report. This make the LTIF command a fairly powerful reporting tool, as it is able to combine token and user information in a useful manner. For instance, this command can be used to report a list of users who have tokens expiring in 45 days. Because user information can be included, the report could be used to generate a mailing list or even to automatically notify users via email.

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

If Security Domain is not provided, then the token will be searched in the default security domain (System Domain) created during Authentication Manager installation. Token searches automatically include all sub domains.

If an Identity Source is not provided, then listings containing User information (outputOption 10 and higher) will be listed for all Identity Sources. If an Identity Source is provided then listings containing User information will be limited to user existing in that Identity Source. To limit User information to the Default Identity Source (usually Internal Database) it must be explicitly declared. When an Identity Sources is declared, it acts like a filter when listing User information for assigned tokens.

An IdentitySource must be provided if the SecurityDomain given in the input file is mapped to a different realm other than the default created during installation. In this case, listings for reports containing User information (outputOption 10 and higher) the listings will be limited to the specified Identity Source.

Select the CompareField and the CompareType values from the following. Supply a value for CompareValue of the type listed in the value column. The default values for CompareField, CompareType, and CompareValue are 0, 0, and 0. If either CompareField or CompareType is 0 or left blank, both will take the value 0.

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

Token Listed	Field	Type	Value
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
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

* Token type values:

0 RSA SecurID Standard Card

1	RSA SecurID PINPAD Card
2	RSA SecurID Key Fob
3	RSA SecurID Watch
4	RSA SecurID Software Token (formerly SoftID)
5	RSA SecurID Smartcard
6	RSA SecurID Modem
7	RSA SecurID Crypto
8	RSA SecurID Proteus
9	RSA SecurID USBCOSMO (SID800)
10	RSA SecurID Flextoken

** Software Token Device Type

Can be specified as follows:

Empty All Software Token Devices Type qualify.

“None”(no quotes) Software Tokens with no device type specification qualify.

FamilyKeySoftware 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>will filter software token device types that match name and version.

Examples:

BlackBerry(any version)

BlackBerry**:**3.0(version 3.0 only)

BlackBerry **:** 3.0(same as proceeding example)

Web SDK **:**(any version)

Case is significant for all name matches.

The **OutputOption** may contain the following 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 will be listed only if the selected token is assigned.

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

where:

-31 produce a listing of 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 produce a listing of token serial numbers and CT-KIP download URLs. (Only lists data for CT-KIP capable tokens.

-11 produce a listing of token serial numbers and CT-KIP activation codes. (Only lists data for CT-KIP capable tokens.

-1 produce a listing of token serial numbers only (no data)

0 list token information

10 list user information

11 append ldap source to option 10

12 append remote alias to option 10

13 append profile name to option 10

14 option 10, 11, 12, and 13 combined

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 option 20, 21, 22, and 23 combined

25 token information and ldap source

26 token information and remote alias

27 token information and profile name

H | h write a header record in the output file

A | a open results in append mode.

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

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

[0 | 1 | 2 . . . | 11]

Where:

no additional data listed

include token extension data

include user extension data

option 1 and 2

include group membership group names

option 1 and 4

option 2 and 4

option 3 and 4

include group membership field values

option 1 and 8

option 2 and 8

option 3 and 8

Because these options may 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 will be preceded by *tagBTED* for token extension data and *tagBUED* for user extension data. The tag will be 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 will be 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 may appear more than once. When appropriate, the special identifier tag names will also appear. Header records for the ListTokenInfoByField are as follow:

The following field identifiers will be included in the heading, 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.

The MiscVariable is used to fix up extension keys and/or data that contains commas. Some users have either intentionally or unintentionally entered commas in their extension data keys or data. Because the retrieval of the extension data and keys is returned as a comma separated list, what is the key and what is the data can be next to impossible to figure out. If you are requesting extension data in this list, AMBA attempts to fix up key/data pairs that contain more than one comma. It does so by parsing the results, building 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. Once a key if 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 will be valid as either the first portion only, or any concatenated portion. In any case, MiscVariable may be used to control the substitution. MiscVariable may 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

If the MiscVariable is either absent or empty, the default action is to replace extra commas by semi-colon.

List Token Info For Token Serial

This command will produce token information and user information if the token is assigned. This command duplicates the List Token Info by Field command (above) in every respect except one. The difference is that this command will supply the information for a single supplied token serial.

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

Consult the List Token Info by Field command (above) for the definition of the optional fields, usage and output formatting options.

List Token Summary Report

This command will produce a token summary report. The report consists of an enumeration of each token type in the database and a count of the number of tokens of that type. It will then enumerate the Soft Token Device Types. Both listings will only list an item if and only if one or more items are declared in the database.

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

MiscVariable is used to select token state:

1 - Report on assigned tokens only

2 - Report on unassigned tokens only

3 - Report on all tokens

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

In addition to empty, there is only one outputOption value that this command will accept. That is a | A which will instruct AMBA to create a new file if one does not exist or append the results to a previous operation if a file does exist.

8 Multiple Token Actions

Overview

The multiple token action commands 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, but they are outweighed by the amount of effort that can be saved.

If a database is properly populated with extension data, the output of these commands can be used for automated notification. Just as with the list commands, the output of these commands is a comma-separated variable list that can easily 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 will be the single word ‘Done’ followed by a newline sequence. In this case, a report that generated no output will consist of a single line beginning with the word “Done.”

Multiple Softtoken Deployment

Deploys software tokens in sdtid files which in turn will be stored in zip format. Specifies a range of software tokens and deploys them to assigned users. Prior to calling this function, all users must be in the RSA Authentication Manager database, and all software tokens included in the range must be assigned to a user. The tokens can be deployed to users as specified by serial number, or default login. The following table lists acceptable values for use with rangeMode, the affect each value has on subsequent parameters, and the action taken when the function is called.

Action	**MSD**
Required Fields	rangeMode
Optional Fields	startRange, endRange, password, filename, copyProtect, overOption, IdentitySource, Securitydomain, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, NicknameIsCtkipCode, DeviceserialIsCtkipCode

Note: The tokens in a given range should be of same device type.

In RSA Authentication Manager, each **sdtid** file can have a max of 100 tokens. For example, if the range provided is around 1000, then 10 sdtid files will be available (1 for each 100 tokens) in the Zip file created.

For rangeMode 0, 1 and 2, the identity source option is only used to validate the security domain provided. IdentitySource needs to be provided, if the SecurityDomain given in the input file is mapped to a different realm other than the default created during installation. For these rangeModes, tokens will be fetched based on the SecurityDomain.

The **-g** command line option is required to instruct AMBA to generate software token database files. Optionally, the -gdir command line option may be used to place any generated files in the specified directory. The --ctkip and -g options are mutually exclusive

The **--ctkip** command line option is required to instruct AMBA to generate CT-KIP credentials for qualifying tokens. The --ctkip and -g options are mutually exclusive. Also, the set soft token device type command (sstdt) can be used to force a specific device type for credential generation.

If the **-g** and --ctkip options are not used, the MSD command will not build any output files or generate any CT-KIP credentials. Additionally, if the **-g** command line option is not used, any -gdir command line option will be ignored.

RangeMode	StartRange	EndRange	Action Taken
0	TokSerial	Ignored	Deploys one assigned software token specified by the serial number in the startRange parameter.
1	Ignored	Ignored	Deploys all assigned software tokens.
2	TokSerial	TokSerial	Deploys all assigned software tokens within the specified range of serial numbers.
3	DefLogin	Ignored	Deploys all software tokens that are assigned to a user with the default login specified in startRange.
4	DefLogin	DefLogin	Deploys all software tokens that are assigned to a range of users specified by a default login provided with startRange and endRange.

RangeMode

Specifies criteria used to deploy assigned software tokens either by serial number or user default login. See the table above for details.

startRange

The beginning software token serial number or user default login in a range. If rangeMode is set to 1, this argument is ignored.

endRange

The ending software token serial number or user default login in a range. If rangeMode is set to 0, 1, or 3, this argument is ignored.

password

Specifies that a password be provided by an administrator in order to access an RSA SecurID software token sdtid file. If there is no password associated with the file, an empty string may be passed.

fileName

Specifies the name of the output ZIP file. If left empty, the fileName defaults to **softTokenDeployment.zip**.

copyProtect

Specifies whether the copy protection is enabled for the software tokens in the ZIP file. If copy protection is enabled, the Software Token record cannot be removed from the directory in which it is installed on a user’s computer. If 0, copy protection is disabled; if any other value, copy protection is enabled.

overOption

If any value other than 0, the file is overwritten; If 0 is used and if the file already exists, the content (.sdtid files) will be appended to the existing Zip file - otherwise a new Zip file will be created with the name provided.

IdentitySource

If rangeMode is set to 0, 1, or 2, this argument is only used to validate the security domain provided. For rangeMode 3 and 4, the range of users provided will be searched in the given identity source. If not provided, then the users will be searched in the identity source mapped to the default security domain, for example, the **Internal DataBase** created during installation.

SecurityDomain

The range of tokens provided will be searched in the given security domain. If not provided, then the tokens will be searched in the default security domain, for example, the **System Domain** created during installation.

Software Token Device Type Attributes

A specific Software Token Device Type may be linked to a software token through the security console, the API or the AMBA SSTDT command. Software Token Device Types contain various sets of attributes which are basically name/value pairs. Default values may be assigned to these attributes through the security console, the APIs or AMBA. Use the variables declared in the following table to assign values to the attributes for specific tokens.

Software Token Device Type Attributes	AMBA Variable 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

Software Token Device Type Attributes	AMBA Variable Name	Permitted Values
TOOLBAR_SITEURL2	SiteRUL2	0, value, empty
TOOLBAR_SITEURL3	SiteURL3	0, value, empty

Attribute values:

-2 - copy the TokSerial to the attribute value field.

-1 - copy the DefLogin to the attribute value field

0 - force 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

For example:

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 above input will create two Zip files, SerialRange.zip and UserRange.zip.

The first input line will create a file named SerialRange.zip containing the range of tokens from 000027050103 to 000027050109 in the form of an sdtid file.

The second input line will create a file named UserRange.zip containing the range of tokens associated with the users from user1 to user 50 in the form of an sdtid file.

The last line will append the token 000027050110 in the form of an sdtid file to the already existing file SerialRange.zip.

Multiple Token Assignment

This command 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 will be assigned. The type of token to be assigned is determined by an input parameter. When the token is assigned, its PIN is cleared and the token is disabled. A listing will be generated. The list content is determined by input parameters. The listing will indicate the token serial number assigned and the user that it’s assigned to. The listing can be useful for automated notification or mailing label generation.

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

Note: If software tokens are being assigned, the -g and --gdir command line options may be used to instruct AMBA to generate software token database files and place them in the specified directory. The filename defaults to **softTokenDeployment.zip**. Consult the ctkip command line option and the sstdt command for ctkip credentials generation options.

Input field definitions:

CompareField - This parameter is used to indicate what type of token to assign.

-1 Any unassigned token (default if empty)

0 RSA SecurID Standard Card

1 RSA SecurID PINPAD Card

2 RSA SecurID Key Fob

3 RSA SecurID Watch

4 RSA SecurID Software Token (formerly SoftID)

5 RSA SecurID Smartcard

6 RSA SecurID Modem

7 RSA SecurID Crypto

8 RSA SecurID Proteus

9 RSA SecurID USBCOSMO (SID800)

10 RSA 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 will run until the last available token has been assigned. At that point, the process will terminate and an appropriate entry will be recorded in the log.

OutputOption - This parameter is used to determine the information that will be 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 comma separated variable format (CSV). Most scripting languages as well as document and spreadsheet editors can parse these files and generate the desired end format.

0 - tokSerial, defLogin, FirstName, LastName (default if empty)

1 - Option 0 + ldap source

2 - Option 0 + remote alias

3 - Option 0 + profile name

4 - Option 0 + 1 + 2 + 3

10 - tokSerial, <userInfo>

11 - Option 10 + 1

12 - Option 10 + 2

13 - Option 10 + 3

14 - Option 10 + 1 + 2 + 3

h | H - output header line

r | R - Report only

a | A - Open results in append mode.

User information:

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

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

ExtnDataOption - This parameter is used to indicate whether or not extension data should be included in the output report.

0 - No additional output

2 - Include user extension data

4 - Include group membership group names

6 - Option 2 plus option 4

8 - Include group membership group fields

10 - Option 2 plus option 8

Because these options may 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 will be preceded by *tagBTED* for token extension data and *tagBUED* for user extension data. The tag will be 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 will not appear. If group membership data appears in the output, the first group membership name or the first group membership field will be preceded by the tag *tagBGM*. If there are no group membership values, the tag will not appear.

The following field identifiers will be 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 has been requested, appropriate field names are written for each option. The field names will appear only once, even though the actual data may appear more than once. When appropriate, the special identifier tag names will also appear.

MiscVariable - This parameter is used to control extension data comma replacement. Many users, either intentionally or unintentionally, have included commas in extension data. The Authentication Manager custom API, (used by AMBA 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 parameter can help eliminate these issues, by replacing or deleting the extension data commas. The replacement, if requested, only affects the generated report and is never saved back in the database.

<char> - character to replace imbedded 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.

TokEnabled - This parameter is used to control the enabled/disabled state of the assigned token.

0 - Disable the assigned token (default)

1 - Enable the assigned token

SetPin - This parameter is used to initialize the PIN of the newly assigned token.

0 - Clear PIN. (Default) This action will automatically put the token in new PIN mode.

? - Any other value will be will be used as the new pin.

Note: Assigning a literal value to SetPin will set all newly assigned tokens to this PIN.

PinMode - This parameter is used to control the new PIN mode of the newly assigned token.

0 - No action is taken (Default)

1 - Set the newly assigned token to new PIN mode.

PinType - This parameter is used to specify whether or not a token requires a pin.

Passcode - The token is a passcode token (requires a pin)

Tokencode - The token is a tokencode only token (no PIN required).

SoftIDParams - This field is used when a softID token seed record file is being created. When used, this field must contain three decimal digits (to maintain consistency with ACEBulkAdmin utility). The three decimal digits control the following seed file generation characteristics:

First digit:

0 - required but ignored

Copy Protection Flag 0 or 1:

0-Copy protection off

1 -Copy protection on

Password usage and Interpretation method 0, 1, 2, or 3:

0- No password

1- Static password (See SoftIDPW below)

2- Default login

3- Default login appended to static password

SoftIDPW - This field is used when a softID token seed record field is being created. When used, this field supplies a password to be used for the seed file encryption. This field supplies a password to be used for the seed file encryption when **SoftIDParams** specifies a static password.

IdentitySource - The range of users will be searched in the given identity source. If not provided, then the users will be searched in the identity source mapped to the default security domain, for example, the **Internal DataBase** created during installation.

SecurityDomain - The range of tokens will be searched in the given security domain. If not provided, then the tokens will be searched in the default security domain, for example, the **System Domain** created during installation.

Software Token Device Type Attributes

A specific Software Token Device Type may be linked to a software token through the security console, the API or the AMBA SSTDT command. Software Token Device Types contain various sets of attributes which are basically name/value pairs. Default values may be assigned to these attributes through the security console, the APIs or AMBA. Use the variables declared in the following table to assign values to the attributes for specific tokens.

Software Token Device Type Attributes	AMBA Variable 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	SiteRUL2	0, value, empty
TOOLBAR_SITEURL3	SiteURL3	0, value, empty

Attribute values:

-2 - copy the TokSerial to the attribute value field.

-1 - copy the DefLogin to the attribute value field

0 - force 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

Additionally, to improve efficiency the MTA command processes token assignments in groups of 500. A side effect of this is that only one log entry is made at the completion of the command. 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.

Multiple Token Disable/Rescind

This command is used to scan the database for assigned tokens that have not been used (logged on) 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 will indicate the token serial number being disabled/rescinded and the user that it's assigned to. Additional listing information is available as noted below. The listing can be useful for automated notification or mailing label generation.

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

Note: The identity source provided is only used to validate the security domain provided. IdentitySource needs to be provided if the SecurityDomain given in the input file is mapped to a different realm other than the default created during installation. Processing of this command is purely driven by the Security Domain value.

Input field definitions:

CompareField - This optional parameter is used to indicate what type of token to scan for.

-1	Any assigned token (default if empty)
0	RSA SecurID Standard Card
1	RSA SecurID PINPAD Card
2	RSA SecurID Key Fob
3	RSA SecurID Watch
4	RSA SecurID Software Token (formerly SoftID)
5	RSA SecurID Smartcard
6	RSA SecurID Modem
7	RSA SecurID Crypto
8	RSA SecurID Proteus
9	RSA SecurID USBCOSMO (SID800)
10	RSA SecurID Flextoken

CompareType - This parameter designates the type of action 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.

CompareValue - This optional parameter is used to set the cutoff date to determine inactivity.

Valid entries:

0 or empty - Cutoff date is (current date - 90) days

A number from 1 through 365 - Cutoff date is set to (current date - number).

A date of in the following format, MM/dd/yyyy - The date must be greater than or equal to the (current date - 365 days) and less than current date. The cutoff date will be set to this date.

OutputOption - This parameter is used to determine the information that will be included in the output report for each token that is disabled. 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 comma separated variable format (CSV). Most scripting languages as well as document and spreadsheet editors can parse these files and generate the desired end format.

0 - tokSerial, defLogin, FirstName, LastName (default if empty)

1 - Option 0 + ldap source

2 - Option 0 + remote alias

3 - Option 0 + profile name

4 - Option 0 + 1 + 2 + 3

10 - tokSerial, <userInfo>

11 - Option 10 + 1

12 - Option 10 + 2

13 - Option 10 + 3

14 - Option 10 + 1 + 2 + 3

h | H - output header line

r | R - Report only

a | A - Open results in append mode.

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

ExtnDataOption - This parameter is used to indicate whether or not extension data should be included in the output report.

no additional data listed (default if empty)

include token extension data

include user extension data

option 1 and 2

include group membership group names

option 1 and 4

option 2 and 4

option 3 and 4

include group membership field values

option 1 and 8

option 2 and 8

option 3 and 8

Because these options may 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 will be preceded by *tagBTED* for token extension data and *tagBUED* for user extension data. The tag will be 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 will not appear. If group membership data appears in the output, the first group membership name or the first group membership field will be 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 may appear more than once. When appropriate, the special identifier tag names will also appear.

MiscVariable - This parameter is used to control extension data comma replacement. Many users, either intentionally or unintentionally, have included commas in extension data. The Authentication Manager custom API, (used by AMBA 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 parameter can help eliminate these issues, by replacing or deleting the extension data commas. The replacement, if requested, only affects the generated report and is never saved back in the database.

<char> - character to replace imbedded 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 imbedded commas with the semi-colon (;) character.

IdentitySource - It is only used to validate the security domain provided. If not provided, then the identity source mapped to the default security domain, for example, the **Internal DataBase** will be used.

SecurityDomain - The range of tokens will be searched in the given security domain. If not provided, then the tokens will be searched in the default security domain, for example, the **System Domain** created during installation.

Additionally, to improve efficiency the MTD command processes token assignments in groups of 500. A side effect of this is that only one log entry is made at the completion of the command. 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.

Multiple Token Replacement

This command is used to scan the database for tokens that have expired, are about to expire or both. 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 an input parameter. When the token is assigned, its PIN is cleared and the token is disabled. A listing will be generated. The list content is determined by input parameters. The listing will indicate the token serial number assigned and the user that it's assigned to. The listing can be useful for automated notification or mailing label generation.

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

Note:

The -g command line option is required to instruct AMBA to generate software token database files. Optionally, the -gdir command line option may be used to place any generated files in the specified directory. The filename defaults to **softTokenDeployment.zip**.

The --ctkip command line option is required to instruct AMBA to generate CT-KIP credentials for qualifying tokens. The --ctkip and -g options are mutually exclusive. Also, the set soft token device type command (sstdt) can be used to force a specific device type for credential generation.

If the -g and --ctkip options are not used, the MSD command will assign tokens but will not build any output files or generate any CT-KIP credentials. Additionally, if the -g command line option is not used, any -gdir command line option will be ignored.

Identity source provided is only used to validate the security domain provided. IdentitySource needs to be provided if the SecurityDomain given in the input file is mapped to a different realm other than the default created during installation. Processing of this command is purely driven by the Security Domain value.

Input field definitions:

CompareField - This parameter is used to indicate what type of token to assign.

-2 Same as token being replaced

-1 Any unassigned token (default if empty)

0 RSA SecurID Standard Card

1 RSA SecurID PINPAD Card

2 RSA SecurID Key Fob

3 RSA SecurID Watch

4 RSA SecurID Software Token (formerly SoftID)

5 RSA SecurID Smartcard

6 RSA SecurID Modem

7 RSA SecurID Crypto

8 RSA SecurID Proteus

9 RSA SecurID USBCOSMO (SID800)

10 RSA SecurID Flextoken

For CompareField values equal to -1 or -2, **any** expired or about to expire assigned token will be assigned a replacement token. For all other values of CompareField, only assigned tokens who's type matches the CompareField type will be assigned a replacement token.

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 will run until the last available token has been assigned. At that point, the process will terminate and an appropriate entry will be recorded in the log.

If software tokens are being assigned, the -g and -gdir command line options maybe used to instruct AMBA to generate software token database files and place them in the specified directory.

CompareType - This parameter 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 (time in which token will expire)

0 - (the default)

days from current date

MM/dd/yyyy

TokenSerial -

0 - unconditional replacement (default)

1 - replace if expired token is the only assigned token

TokEnabled - This parameter is used to control the enabled/disabled state of the assigned token.

0 - Disable the assigned token (default)

1 - Enable the assigned token

PinMode - This parameter is used to control the new PIN mode of the newly assigned token. If PinMode is 0 or not set at all, new PIN mode will be reset (forced off) for all newly assigned tokens. This will cause each newly assigned token to inherit the PIN and the new PIN mode state of the token it is replacing. If PinMode is set to 1, each newly assigned token will be placed in new PIN mode, its PIN will be cleared and will NOT inherit the PIN of the token being replaced.

0 - Replacement token will inherit the PIN and new PIN mode state of the token it is replacing

1 - Clears the PIN of the newly assigned token and forces new PIN mode on.

PinType - This parameter is used to specify whether or not a token requires a pin.

Passcode - The token is a passcode token (requires a pin)

Tokencode - The token is a tokencode only token (no PIN required).

OutputOption - This parameter is used to determine the information that will be 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 comma separated variable format (CSV). Most scripting languages as well as document and spreadsheet editors can parse these files and generate the desired end format.

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)

0 - Base Information only (default if empty)

1 - Option 0 + ldap source

2 - Option 0 + remote alias

3 - Option 0 + profile name

4 - Option 0 + 1 + 2 + 3

10 - Base Information + <userInfo>

11 - Option 10 + 1

12 - Option 10 + 2

13 - Option 10 + 3

14 - Option 10 + 1 + 2 + 3

h | H - output header line

r | R - Report only

a | A - Open results in append mode.

u | U - Change the file name to <user ID>.sdtid t | T - Change the file name to <token serial.sdtid

e | E - Extract the sdtid files from the zip file.

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

The default output for mtr commands is a ZIP file titled SoftwareTokens.zip. The contents of the zip file are sdtid files with the default name of <user ID>_<Token Serial>_.sdtid. If the extract files option is included (e | E), the SoftwareTokens.zip file will be unzipped into the same directory and the SoftwareTokens.zip file will be deleted. If the extract files option is included, one of the file naming options may also be included. The u | U option will change the file name to <user ID>.sdtid and the t | T option will change the file name to <token serial>.sdtid. If the extract files option is not included, the file naming options are ignored.

ExtnDataOption - This parameter is used to indicate whether or not extension data should be included in the output report.

0 - No additional output (default if empty)

2 - Include user extension data

4 - Include group membership group names

6 - Option 2 plus option 4

8 - Include group membership group fields

10 - Option 2 plus option 8

MiscVariable - This parameter is used to control extension data comma replacement. Many users, either intentionally or unintentionally, have included commas in extension data. The Authentication Manager custom API (used by AMBA 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 parameter can help eliminate these issues, by replacing or deleting the extension data commas. The replacement, if requested, only affects the generated report and is never saved back in the database.

<char> - character to replace imbedded 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 imbedded commas with the semi-colon (;) character.

MinTokenLife - This field is optional and is used to guarantee that a replacement token will have a minimum number of days before its expiry date. The search for a token looks for tokens with an expiry date greater than the cutoff date (compareValue). Adding a number of days with the MinTokenLife variable will adjust the search to tokens with an expiry date greater than the cutoff date + MinTokenLife. If MinTokenLife is not supplied, it defaults to 90. MinTokenLife may be supplied in units of days or as an explicit date (MM/dd/yyyy) and it must result in a date greater than the cutoff date.

SoftIDParams - This field is used by the add-token commands when a softID token seed record file is being created. When used, this field must contain three decimal digits (to maintain consistency with ACEBulkAdmin utility). The three decimal digits control the following seed file generation characteristics:

First digit:

0 - required but ignored

Copy Protection Flag 0 or 1:

0-Copy protection off

1 -Copy protection on

Password usage and Interpretation method 0, 1, 2, or 3:

0- No password

1- Static password (See SoftIDPW below)

2- Default login

3- Default login appended to static password

SoftIDPW- This field is used by the add-token commands when a softID token seed record field is being created. When used, this field supplies a password to be used for the seed file encryption when **SoftIDParams** specifies a static password

IdentitySource - It is only used to validate the security domain provided. If not provided, then the identity source mapped to the default security domain, for example, the **Internal DataBase** will be used.

SecurityDomain - The range of tokens will be searched in the given security domain. If not provided, then the tokens will be searched in the default security domain, for example, the **System Domain** created during installation.

Software Token Device Type Attributes

A specific Software Token Device Type may be linked to a software token through the security console, the API or the AMBA SSTDT command. Software Token Device Types contain various sets of attributes which are basically name/value pairs. Default values may be assigned to these attributes through the security console, the APIs or AMBA. Use the variables declared in the following table to assign values to the attributes for specific tokens.

Software Token Device Type Attributes	AMBA Variable 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	SiteRUL2	0, value, empty
TOOLBAR_SITEURL3	SiteURL3	0, value, empty

Attribute values:

-2 - copy the TokSerial to the attribute value field.

-1 - copy the DefLogin to the attribute value field

0 - force 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

Additionally, to improve efficiency the MTR command processes token assignments in groups of 500. A side effect of this is that only one log entry is made at the completion of the command. 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.

9 General Functions

Change Date Format

The change date format command may be used anywhere in the input file and as many times as desired. This command is used to override the default date format which is normally empty. This command will also replace any date format created with the datefmt command line and INI file options. If the MiscVariable field is missing or empty the date format will be cleared and returned to its default state. The MiscVariable field is used to provide a Java SimpleDateFormat string to use. No syntax checking is performed on this value so errors may occur when the format is first used or incorrect results may be encountered. The date format string is used to format List command (LUIF, LUI, LTIF, LTI) Java Date objects. Do not use quotes in these strings unless you want them to appear in the output.

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

Example Date Formats

yyyy/MM/dd

yyyy-MM-DD HH:mm:ss

The following pattern letters are defined. Letter case is significant. All other characters from 'A' to 'Z' and from 'a' to 'z' are reserved):

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

Letter	Date or Time Component	Presentation	Examples
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
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

The change input format command may be used anywhere in the input file and as many times as desired. This command is used to override the default input file format and simplify the input file build process. It is also used to dynamically redefine the input file format at run time. This command consists of ‘CIF’ in the action field followed by one or more field labels. The CIF command is case insensitive. The number and order of field names is arbitrary.

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, SoftIDParams, SoftIDPW, RemoteAlias, RealmName, CompareField, CompareType, CompareValue, OutputOption ,ExtnDataOption ,MiscVariable, ProfileName ,TokenSerial ,rangeMode ,startRange ,endRange ,password, filename ,copyProtect ,overOption, Email, CertDN, Key, KeyType, SecurityDomainName, ParentDomainName,SecurityDomainDescription, SecurityDomainCreatedBy, PolicyType1, PolicyName1, PolicyType2, PolicyName2, PolicyType3, PolicyName3, PolicyType4, PolicyName4, PolicyType5, PolicyName5, Nickname, DeviceSerialNumber, SiteFile, SiteURL1, SiteURL2, SiteURL3, NicknameIsCtkipCode, DeviceserialIsCtkipCode, DestinationSecurityDomain, EnableFlag, OTPLength, OTPInterval, OTPAlgorithm, PinAdded, ForceGroupSearch

This command is intended to simplify the input file build process by allowing the user to define command lines unique to each user’s requirement. An example of this would be to define the first set of input as add group (AG), followed by add user to group, followed by add user with token. A file of this type might look something like the following:

```
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 command fails for any reason, all input lines are logged and ignored until a valid CIF command is processed or the end of file, which ever occurs first.

Change Results File Name

Previously, the results file name could be changed on the command line and the file name was in effect for the entire session. This command may be 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. Therefore, this command may be used between multiple listing commands in one session. This will prevent one list command from overwriting the results of a previous list command in any one session. The result file name is supplied in the MiscVariable. This variable may include a drive, path and extension, where applicable. Leaving MiscVariable empty resets the result file name back to its default.

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

Quit

The quit command is used to terminate the standard input file (STDIN). It is ignored if present in a disk input file.

Action	**QUIT**
Required Fields	None
Optional Fields	None

A

Automatic Notification

The Automatic Notification System may be used to automatically send email messages when provisioning software tokens. This action is triggered when the “deliveryMethod” field of a token command is set to either ‘SMTP” or “SMTP2” and AMBA has been initiated with either the “-g” or “—ctkip” command line option. If the token command is successful, then the appropriate email will be sent.

By default, the principal’s account email address will be used however this can be overridden by including the “destinationAddress” variable with the token command and setting it to the desired email address.

AMBA 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 template files provided with AMBA. The template files are provided for users who wish to customize the email text and format. 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 may 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 may 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 "">	(ctkip activation code)
<!ENTITY url "">	(ctkip activation url)

Once an ENTITY is declared in DOCTYPE, its “place holder” may 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 test it will be replaced by the users default logon when the email is generated.

It should be noted that not all ENTITY names can be used in all templates. For example, PIN is applicable to ON Demand notices only. Although PIN would not produce an error on a token provisioning notice, it will be empty.

There are two deliveryMethods for SDTID files (SMTP & SMTP2). Both methods will generate an email with the token SDTID file as an attachment however SMTP2 will generate a followup 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.

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 will be used for the password notification. For example:

custom_template.xml(token attachment template)

custom_templatepw.xml(password notification template)

Name with no suffix

custom_template(token attachment template)

custom_templatepw(password notification 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 a SDTID file named c:\temp\userid_tokenserial.sdtid will be named c__temp__userid_tokenserial.sdtid as the email attachment name.

Currently, for Automatic Notification the 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)**.

If multiple instances are configured on this server you must add the InstanceName variable to the token command and specify which instance to use for the mail server. If only one instance has been declared then the InstanceName variable is not necessary.

B Troubleshooting

java.lang.ClassNotFoundException:

If the log displays the following error message, it is almost always caused by an improper AMBA 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
```

AMBA requires a valid Enterprise license file or a standalone AMBA license. For instructions, see Install AMBA on page 12.

In some instances, it may 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 RSA Authentication Manager.

Flush the Cache

Flush the cache to remove old information from memory. When you flush the cache, each selected 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**.

C Command Table

This appendix contains a reference table that describes AMBA commands and also provides information on AceBulkAdmin fields that are not supported in AMBA.

Changes from AceBulkAdmin

Some of the fields present in AceBulkAdmin might be considered as Unsupported Fields in AMBA. For example, while adding a user **ExtnKey, ExtnData** fields are not taken into consideration, and if any values are provided, they are ignored and written to the unsupported options log.

Each run of AMBA produces an Unsupported Options file called **AMBulkAdminUnsupportedOptions.txt**, in the current directory, containing the list of all unsupported fields for each command entered in the input file. This file contains line number, action field and the unsupported fields found in each line.

The following example is a listing from an **AMBulkAdminUnsupportedOptions.txt** file. If the Input File contained the following information:

```
action,deflogin,lastname,UserPwd,ExtnKey,ExtnData
au,User1,User1,Secret1!,Key1,Data1
au,User2,User2,Secret2$,,
```

The **AMBulkAdminUnsupportedOptions.txt** file generated would contain the following:

```
// Unsupported properties for each input line
Line 2 : Action - au : Unsupported Fields - ExtnData, ExtnKey
```

For information on field and functionality changes, see Alphabetical Command Table on page 134.

Alphabetical Command Table

The following table lists the fields and functionality in AMBA.

Note: Almost all commands in AMBA require either **Security Domain** or **Identity Source** as input fields. If their values are not provided, then the default values of **SystemDomain** and **Internal Database** are used.

Command Type	CMD	Description	Field and Functionality Changes in AMBA
Add commands	au	Add User Add a new user, and optionally, add a user to an existing group.	**UserPwd** is an additional, optional field. ClntName, ClntDefLogin, ClntDefShell,ExtnKey, ExtnData are unsupported.
Add commands	aug	Add User to Group	None
Add commands	aup	Add User and Password Add a user with a password, and optionally, add a user to an existing group.	**UserPwd** is an additional, optional field. CreatePin, ClntName, ClntDefLogin, ClntDefShell, ExtnKey, ExtnData are not supported.
Add commands	aur	Add User Remote	ClntName, ClntDefLogin, ClntDefShell, ExtnKey, ExtnData are not supported.
Add commands	aut	Add User and Token Add a new user with a token, and optionally, add a user to an existing group	**UserPwd** is an additional, optional field. **CreatePin** field is unsupported G,g, **-1** values for **SetPin** field are unsupported. Encryption key type value in **SoftIdParams** field is unsupported. **ClntName, ClntDefLogin, ClntDefShell, ExtnKey, ExtnData** are unsupported now.
Add commands	auta	Add User and Token Automatic Add a new user and assign an available token from the system.	
Add commands	agc	Add Group to Client Enables a group on a client.	@ (or any group site separator) cannot be used as a separator in the **GrpName** field. The **Security Domain** field should be used to input **Security Domain Name(Site Name)**.

Command Type	CMD	Description	Field and Functionality Changes in AMBA
Add commands	ap	Assign Profile Assign a profile to a user.	None.
Addcommands	asd	Add Security Domain	SecurityDomainDescription, SecurityDomainCreatedBy, PolicyType1,PolicyName1, PolicyType2,PolicyName2, PolicyType3,PolicyName3, PolicyType4,PolicyName4, PolicyType5,PolicyName5,
Change commands	cau	Change or Add User Change or add new user. If deflogin exists, update it. Otherwise add user (au).	**UserPwd** is an additional, optional field. ClntName, ClntDefLogin, ClntDefShell, ExtnKey, ExtnData are unsupported now.
Change commands	caup	Change or Add User and Password Change or add new user with password. If deflogin exists, update user. Otherwise add user with password (aup).	**UserPwd** is an additional, optional field. CreatePin, ClntName, ClntDefLogin, ClntDefShell, ExtnKey, ExtnData are unsupported now.
Change commands	caut	Change or Add User and Token Change or add new user with token. If token already assigned, update user. Otherwise add user with token (aut).	**UserPwd** is an additional, optional field. CreatePin, ClntName, ClntDefLogin, ClntDefShell, ExtnKey, ExtnData are unsupported now. G,g,-1 values for SetPin field is unsupported.
Change commands	cps	Change PIN Status	**G,g,-1** values for **SetPin** field is unsupported.
Change commands	cti	Change Token (Immediate)	None.
Change commands	cts	Change Token Status	None.

Command Type	CMD	Description	Field and Functionality Changes in AMBA
Change commands	ctsx	Change Token Status eXtended	**UserPwd** is an additional, optional field. Static password cannot be enabled without giving a new value in the **SetPin** field.
Change commands	cusd	Change User/Token Security Domain	Move user and optionally tokens to a new security domain
Change commands	cur	Change User Remote	None.
Change commands	eafxd	Set Emergency Access Fixed	None.
Change commands	eaotp	Set Emergency Access OTP	None.
Change commands	eaoff	Set Emergency Access OFF	None.
Change commands	uud	Update User Data	**UserPwd** is an additional, optional field.
Change commands	ctd	Change Token (on First Use of New Token) Change token delayed.	None.
Change commands	ctu	Change Temporary User Change temporary status for a user.	None.
Emergency comamnds	efxd	Set Emergency Access Fixed	
Emergency comamnds	eotp	Set Emergency Access OTP	
Emergency comamnds	eoff	Set Emergency Access OFF	
Set Commands	sstdt	Set Soft Token Device Type Sets a temporary Software Token Device Type	A new command

Command Type	CMD	Description	Field and Functionality Changes in AMBA
Command Line options	<null>	Command-line options Displays input options and its usage.	None.
Command Line options	--debug	Command-line options Debug option will allow validating an input file for required fields without making changes to the database.	"--" instead of "-"
Command Line options	-g	Command-line options To output the software tokens sdtid file.	None.
Command Line options	--gdir<dirname>	Command-line options To store the sdtid files in the dir specified.	"--" instead of "-"
Command Line options	--gtc <tempname>	Command-line options Creates a template file in CSV	"--" instead of "-"
Command Line options	-i <datafile> | stdin	Command-line options For providing the input data file	None.
Command Line options	--ini <inifile>	Command-line options Where <inifile> is the path to the input parameter file	"--" instead of "-"
Command Line options	-m <0 | 1 | 2 | 3>	Command-line options Message logging level	None.

Command Type	CMD	Description	Field and Functionality Changes in AMBA
Command Line options	--newlog	Command-line options Forces AMBA to create a new log instead of appending to the existing log	“--“ instead of “-“
Command Line options	--nolog	Command-line options Turns off all AMBA logging	“--“ instead of “-“
Command Line options	-o <log file> | stderr | stdout	Command-line options Where <log file> is the path to a file for storing the log output	None.
Command Line options	-p <1 | 2 | . . . 3600>	Command-line options Enables the displaying of a “progress report	None.
Command Line options	-r <results file> | stdout | stderr	Command-line options Where <results file> is the path and file name for storing the results of a “List” command.	None.
Command Line options	--rej <command reject file>	Command-line options This is a file containing rejected input records, where <command reject file> is a fully qualified path and file name to be used for this file	“--“ instead of “-“
Command Line options	-v	Command-line options Displays the utility's version number	None.

Command Type	CMD	Description	Field and Functionality Changes in AMBA
Command Line options	--verbose	Command-line options Enable enhanced logging.	“--“ instead of “-“
Command Line options	--userPwd	Command-line options Default password used to create an IMS user.	New option added in AMBA.
Command Line options	-x <0 | 1>	Command-line options Whether datestamp is to be appended to the file names	None.
Command Line options	-a	Command-line options Name of admin User.	New option added in AMBA.
Command Line options	-P	Command-line options Password for specified admin user.	New option added in AMBA.
Command Line options	--lic	Command-line options License file location	New option added in AMBA.
Delete commands	dg	Delete Group	None.
Delete commands	dgc	Delete Group from Client	None.
Delete commands	dt	Delete Token Delete token from User	None.
Delete Commands	dsd	Delete Security Domain	SecurityDomainDescription, SecurityDomainCreatedBy, PolicyType1,PolicyName1, PolicyType2,PolicyName2, PolicyType3,PolicyName3, PolicyType4,PolicyName4, PolicyType5,PolicyName5,

Command Type	CMD	Description	Field and Functionality Changes in AMBA
Delete commands	du	Delete User	None.
Delete commands	dug	Delete User from Group	DefLogin is a required field now instead of GrpDefLogin.
Delete commands	up	Unassign Profile Unassign a profile for a user.	None.
Delete commands	rt	Rescind Token Rescind token	None.
Delete commands	rept	Replace Token Replace token	New command.
Delete commands	repta	Replace Token Automatic	New command.
Delete commands	ut	Unassign Token	None.
General Commands	cif	Change Input Format	None.
General Commands	crfn	Change Results File Name	None.
General Commands	quit	Quit Terminate the **stdin** input file.	None.
List Commands	lsn	List Secondary Nodes for Agent Host	None.
List Commands	ltif	List Token Info by Field	Tokens cannot be assigned to Remote Users. As a result, Results file will not contain Remote alias details for the following output options:12,14,16,22,24,26
List Commands	lti	List Token Info For Token Serial List token information for token serial number	
List Commands	ltsr	List Token Summary Report	Produces an summary report displaying the number of each type of token (when > 0) and the number of each soft token device type (when > 0). Can filter on assignment and securityDomain.

Command Type	CMD	Description	Field and Functionality Changes in AMBA
List Commands	luif	List User Info by Field	Compare Field 4 for search based on default shell is not supported. Compare Type 1 and 2 for search option (9) based on profile is not supported. LDAP data search option (8) will use Last name for searching. For local or remote user search option (5), we will not be supporting the extndata and output options for compare Type 2 - which is remote user, since the system will not have any of those details for a remote user. List user by First Name without a value is not supported.
List Commands	lui	List User Info for User	
Multiple Commands	msd	Multiple Softtoken Deployment	None.
Multiple Commands	mta	Multiple Token Assignment Scan database and assign tokens to qualified users.	None.
Multiple Commands	mtd	Multiple Token Disable/Rescind Scan the database and disable or rescind tokens based on last login date.	None.
Multiple Commands	mtr	Multiple Token Replacement Scan database and assign replacement tokens to qualified users.	None.
On Demand Commands	eoda	Enable OnDemand Authentication	None.

Command Type	CMD	Description	Field and Functionality Changes in AMBA
On Demand Commands	uoda	Update OnDemand Authentication Update On Demand Authentication details	None.
On Demand Commands	doda	Disable OnDemand Authentication	None.

D Sample AMBA Scenarios

The following sample AMBA scenarios show how you could set up CSV input file for the AMBA utility.

You could copy each of these CSV file examples into a text editor, such as Notepad or vim, and save a file with the extension **.csv**. You can then open the file using a spreadsheet program, such as Microsoft Excel, and modify the data.

For more information, see Preparing the Datafile on page 21.

Create New Users, Assign PINs and Groups, and Provision Tokens

This example creates 10 new users, assigns software tokens and system-generated PINs, assigns the users to one of three groups (Sales, IT, or Eng), and provisions the tokens through CT-KIP over SMTP (email).

```
Action,Key,KeyType,MiscVariable,,,,,,,,,,,,,,,
SSTDT,BlackBerry,FamilyKey,3.5,,,,,,,,,,,,,,,
cif,action,LastName,DefLogin,TokEnabled,MiscVariable,FirstName,
Email,MinTokenLife,SetPin,PinType,GrpName,UserPwd,IdentitySourc
e,SecurityDomain,Nickname,DeviceSerialNumber,DeliveryMethod,Tem
plateFile
AUTA,autaln,autauser,1,4,autafn,autaln@rsa.com,90,1234,Passcode
,Sales,password1!,Internal
Database,SystemDomain,,00-11-22,SMTP,CTKIPMod.xml,
AUTA,autaln1,autauser1,1,4,autafn1,autaln1@rsa.com,90,1234,Pass
code,Sales,password1!,Internal
Database,SystemDomain,,00-11-22,SMTP,CTKIPMod.xml,
AUTA,autaln2,autauser2,1,4,autafn2,autaln2@rsa.com,90,1234,Pass
code,Sales,password1!,Internal
Database,SystemDomain,,00-11-22,SMTP,CTKIPMod.xml,
AUTA,autaln3,autauser3,1,4,autafn3,autaln3@rsa.com,90,1234,Pass
code,IT,password1!,Internal
Database,SystemDomain,,00-11-22,SMTP,CTKIPMod.xml,
AUTA,autaln4,autauser4,1,4,autafn4,autaln4@rsa.com,90,1234,Pass
code,IT,password1!,Internal
Database,SystemDomain,,00-11-22,SMTP,CTKIPMod.xml,
AUTA,autaln5,autauser5,1,4,autafn5,autaln5@rsa.com,90,1234,Pass
code,Eng,password1!,Internal
Database,SystemDomain,,00-11-22,SMTP,CTKIPMod.xml,
AUTA,autaln6,autauser6,1,4,autafn6,autaln6@rsa.com,90,1234,Pass
code,Eng,password1!,Internal
Database,SystemDomain,,00-11-22,SMTP,CTKIPMod.xml,
AUTA,autaln7,autauser7,1,4,autafn7,autaln7@rsa.com,90,1234,Pass
code,IT,password1!,Internal
Database,SystemDomain,,00-11-22,SMTP,CTKIPMod.xml,
AUTA,autaln8,autauser8,1,4,autafn8,autaln8@rsa.com,90,1234,Pass
code,IT,password1!,Internal
Database,SystemDomain,,00-11-22,SMTP,CTKIPMod.xml,
AUTA,autaln9,autauser9,1,4,autafn9,autaln9@rsa.com,90,1234,Pass
```

```
code,Eng,password1!,Internal
Database,SystemDomain,,00-11-22,SMTP,CTKIPMod.xml,
```

Replace and Provision Tokens

This example replaces software tokens for 10 users (and the new tokens must have a minimum life of 120 days), enables the tokens, sets the tokens to New PIN mode, and provisions the tokens through a password-protected **SDTID** file.

```
action,TokSerial,MiscVariable,TokEnabled,SetPin,IdentitySource,
SecurityDomain,DeviceSerialNumber,DeliveryMethod,SoftIDPW,SoftI
DParams,MinTokenLife
REPTA,132251663,4,1,C,Internal
Database,SystemDomain,S-111000555,SMTP2,password,1,120
REPTA,132251664,4,1,C,Internal
Database,SystemDomain,S-111000555,SMTP2,password,1,120
REPTA,132251665,4,1,C,Internal
Database,SystemDomain,S-111000555,SMTP2,password,1,120
REPTA,132251666,4,1,C,Internal
Database,SystemDomain,S-111000555,SMTP2,password,1,120
REPTA,132251667,4,1,C,Internal
Database,SystemDomain,S-111000555,SMTP2,password,1,120
REPTA,132251668,4,1,C,Internal
Database,SystemDomain,S-111000555,SMTP2,password,1,120
REPTA,132251669,4,1,C,Internal
Database,SystemDomain,S-111000555,SMTP2,password,1,120
REPTA,132251670,4,1,C,Internal
Database,SystemDomain,S-111000555,SMTP2,password,1,120
REPTA,132251671,4,1,C,Internal
Database,SystemDomain,S-111000555,SMTP2,password,1,120
REPTA,132251672,4,1,C,Internal
Database,SystemDomain,S-111000555,SMTP2,password,1,120
```

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,Iden
titySource,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,
```

Add Agent Hosts

This example adds 10 Agent Hosts (5 are Standard Agents, 5 are Web Agents) that are unrestricted and enabled.

```
action,AgentHostName,AgentHostAddress,AgentHostType,AgentRestriction,EnableFlag,SecurityDomain,Operation
AAH,win-rsa.vcloud.local,198.75.63.22,1,0,TRUE,BusinessDomain,ADD
AAH,win22-rsa.vcloud.local,198.75.63.25,1,0,TRUE,BusinessDomain,ADD
AAH,win23-rsa.vcloud.local,198.75.63.26,1,0,TRUE,BusinessDomain,ADD
AAH,win24-rsa.vcloud.local,198.75.63.27,1,0,TRUE,BusinessDomain,ADD
AAH,win25-rsa.vcloud.local,198.75.63.28,1,0,TRUE,BusinessDomain,ADD
AAH,win26-rsa.vcloud.local,198.75.63.122,2,0,TRUE,BusinessDomain,ADD
AAH,win77-rsa.vcloud.local,198.75.63.132,2,0,TRUE,BusinessDomain,ADD
AAH,win88-rsa.vcloud.local,198.75.63.102,2,0,TRUE,BusinessDomain,ADD
AAH,win99-rsa.vcloud.local,198.75.63.112,2,0,TRUE,BusinessDomain,ADD
AAH,win10-rsa.vcloud.local,198.75.63.92,2,0,TRUE,BusinessDomain,ADD
```

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 SMTP (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
```

Change User Login from “firstname” to “firstinitial.lastname”

This example change the user login for 10 users from “firstname” to “firstinitial.lastname.”

```
Action,DefLogin,LastName,FirstName,IdentitySource,SecurityDomai
n
CAU,autauser,cautlname,cautfname,Internal Database,SystemDomain
CAU,autauser1,cautlname1,cautfname1,Internal
Database,SystemDomain
CAU,autauser2,cautlname2,cautfname2,Internal
Database,SystemDomain
CAU,autauser3,cautlname3,cautfname3,Internal
Database,SystemDomain
CAU,autauser4,cautlname4,cautfname4,Internal
Database,SystemDomain
CAU,autauser5,cautlname5,cautfname5,Internal
Database,SystemDomain
CAU,autauser6,cautlname6,cautfname6,Internal
Database,SystemDomain
CAU,autauser7,cautlname7,cautfname7,Internal
Database,SystemDomain
CAU,autauser8,cautlname8,cautfname8,Internal
Database,SystemDomain
CAU,autauser9,cautlname9,cautfname9,Internal
Database,SystemDomain
```