RSA SecurID Hardware Token Replacement
Best Practices Guide

Version 1

Contact Information

Go to the RSA corporate web site for regional Customer Support telephone and fax numbers: www.rsa.com.

Trademarks

RSA, the RSA Logo and EMC are either registered trademarks or trademarks of EMC Corporation (“EMC”) 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.rsa.com/legal/trademarks_list.pdf.

License Agreement

The white paper and any part thereof is proprietary and confidential to EMC and is provided only for internal use by licensee. Licensee may make copies only in accordance with such use and with the inclusion of the copyright notice below. The white paper and any copies thereof may not be provided or otherwise made available to any other person.

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

The white paper is subject to update without notice and should not be construed as a commitment by EMC.

Note on Encryption Technologies

The referenced 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 the referenced product.

Distribution

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

Disclaimer

EMC does not make any commitment with respect to the software outside of the applicable license agreement.

EMC believes the information in this publication is accurate as of its publication date. EMC disclaims any obligation to update after the date hereof. The information is subject to update without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED TO SUGGEST BEST PRACTICES, IS PROVIDED "AS IS," AND SHALL NOT BE CONSIDERED PRODUCT DOCUMENTATION OR SPECIFICATIONS UNDER THE TERMS OF ANY LICENSE OR SIMILAR AGREEMENT. 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.

All references to “EMC” shall mean EMC and its direct and indirect wholly-owned subsidiaries, including RSA Security LLC.

November 2011

Critical Sections

Planning Considerations: Page 3

RSA SecurID Token Replacement Approach: Page 7

Testing: Page 9

Restrictions and Limitations: Page 10

Introduction

This guide contains additional guidance for customers pursuing hardware token replacement. This guidance focuses on the planning and implementation approach for replacing a customer's existing RSA SecurID hardware tokens with new RSA SecurID hardware tokens in the customer's environment.

This guide is applicable to customer RSA SecurID implementations of all sizes using RSA Authentication Manager 5.2, 6.1 or 7.1. However, the replacement procedures are generally intended for customers with relatively small user populations. Customers with large user populations may require additional tools and assistance, which is available through RSA Professional Services. If you have this type of user population, contact your RSA Account Manager for information about RSA Professional Services.

Note: This guide does not include guidance on conversion from RSA SecurID hardware tokens to RSA SecurID software tokens. Customers with this requirement should contact their RSA Account Manager about using RSA Professional Service in a separate services engagement.

Planning Considerations

RSA strongly recommends that you carefully plan and test your token replacement strategy before rolling it out within a production RSA SecurID environment due to the potential impacts on end-users and the potential to undermine the security posture of the overall system.

This planning should include a phased token distribution approach. This may include a prioritization of targeted user populations and location of users, as well as communications to end-users to notify them of the plan and to tell them what to expect.

For a small number of token replacements, RSA recommends that you use the Authentication Manager administrative interface to replace tokens one at a time. However, to replace large numbers of hardware tokens, RSA recommends that you use bulk administration tools or scripts. When assigning replacement tokens, RSA recommends that the current PIN be maintained on the replacement token so that the token is not placed in New PIN mode. This simplifies the activation of the new token for the end-user.

RSA strongly recommends that you strengthen your PIN policy, but that you do so under a separate initiative or engagement that does not overlap with the replacement of a user's token. This is less intrusive and less confusing for your end-users.

Note: When replacing tokens, a new token can be associated with an existing token for a user. A token can be assigned for replacement days or weeks before a user actually receives the token, at which point they are instructed to, upon their next authentication, use their existing PIN followed by the tokencode from the new replacement token. Once the user is successfully authenticated with the new token, the old token will be automatically unassigned from their account and the new replacement token activated.

For information about PIN policy best practices, see the *RSA Authentication Manager Security Best Practices Guide*.

Consider the following when planning your RSA SecurID hardware token replacement:

1. Thoroughly review the target RSA SecurID environment to ensure a good understanding of the scope of the token replacement project, as well as possible tools required to complete the tasks.

   If the environment has a documented architecture, as well as documented processes (such as for user enrollments, token renewals / replacements, etc.), use this as the basis of the review. If not, the current method for managing end-user tokens (new or replacement) should be well understood and leveraged to minimize the impact on your environment and end-users.

2. Determine which users in the environment require token replacement. Many environments have multiple token types, but the approach in this guide is for hardware tokens only. The Authentication Manager reporting tools or scripts can help identify these users.

3. Determining which users should receive hardware token replacements will likely be subject to a variety of policy and other business requirements and should therefore be carefully planned out. RSA recommends that users with the most privileges, especially those responsible for administering the Authentication Manager environment or other sensitive systems, be given highest priority.

   It may also be desirable or necessary to further split users into manageable groups especially if tokens are typically distributed regionally.

   Use the following table to assist with the creation of a prioritized token replacement strategy:

	User Grouping	User Classification	Notes
RSA Token Users	RSA Admins	High Privilege	RSA Authentication Manager system administrators with the permissions for system configuration, token distribution, and so on. In Authentication Manager 6.1 these are users assigned one or more administrative “Task Lists”; in Authentication Manager 7.1 these are users assigned at least one “Administrative Role”.
		Privileged	Users with some ability to manage tokens or users, but restricted in some fashion from broader access, for example, help desk personnel
	External Admins	Privileged	Users with administrative permissions to RSA agent protected infrastructure, such as servers, firewalls, switches, VPN concentrators, and so on, but not RSA Authentication Manager administrators
	Agent Protected Application Consumers	Special	Users with access to restricted agents or groups most likely users accessing enterprise applications, such as PCI systems, HR, and so on.
	Token End-Users	Standard	Users that use tokens to authenticate to unrestricted applications or for infrastructure access. These users have no special permissions in RSA Authentication Manager or other systems other than being required to use tokens for authentication.

Note: This table is a general guideline only. You may need to establish other prioritization criteria in accordance with your organization’s business requirements, existing policies and procedures, communication plans or mechanisms and any applicable legal, regulatory and/or compliance regimes.

4. Determine the best RSA SecurID hardware token distribution mechanism. Factors that may affect this decision are:

   - Current distribution methods – Leveraging a current distribution method, which is familiar within an environment, is generally easier to implement and can often avoid confusion.

   - The security of the current distribution method – Ensuring that the proper RSA SecurID token is distributed to the proper user.

   - Location of end-users - Determine whether users are in a single office location or remote offices.

   If tokens must be shipped to remote locations and offices, be careful to use accurate shipping addresses to ensure the correct token is delivered to the correct user at the most secure location possible.

5. Your replacement plan must also take into consideration the number of hardware tokens to be replaced. Organizations with large numbers of tokens to be replaced may not have the capacity to store or distribute all tokens in a single phase. In these situations a phased approach and roll out plan may need to be developed.

6. A communication plan, particularly in large environments, is necessary to inform end-users of the replacement plan and how it will be implemented. This helps to set the correct expectations regarding process and timelines. Every organization has its own employee communications mechanism and standards which should be utilized.

   It is especially important to inform the end-users of how to activate their new tokens when they receive them and the need for them to do so as soon as possible. You should also give guidance to end-users to protect against social engineering or other ways in which fraudsters may attempt to subvert or co-opt the process. Advise end-users to alert the proper authorities of any suspicious behavior. Combining such messaging reinforces the importance of the end-users' role in the process. For guidance on preventing social engineering attacks, see the *RSA Authentication Manager Security Best Practices Guide*.

7. Once the token replacement process is initiated, you should have a plan to:

   - Track the progress of the tokens being replaced. If there is too much time between the delivery of a new token and the activation of the token, then you should contact end-users to ensure replacements have not fallen into the wrong hands. Proper reporting and auditing procedures should be established to assist with managing the process without over complicating it. Proper and clear communication of end-user expectations and responsibilities, as described above, can help ensure the process goes as smoothly and quickly as possible.

   - Remove the old token from the Authentication Manager database once a new token has been successfully activated and the old token is unassigned.

   - Dispose of old tokens. Provide guidance to end-users on what to do with their old tokens once the new ones are activated. Ideally, an organization will have a token disposal procedure in place already. If not, however, consideration should be given to creating one and communicating the details to the end-users stressing the importance of proper disposal.

     The following link is RSA's statement on disposal of RSA SecurID tokens which can be incorporated as part of the customer overall disposal plan: http://www.rsa.com/support/pdfs/Token_Disposal_statement.pdf

RSA SecurID Token Replacement Approach

SecurID Token Replacement Using Administrative Interface

As mentioned earlier in this document, for a relatively low number of token replacements, RSA recommends that you use the Authentication Manager administrative interface to replace tokens one at a time. This is the high-level process for token replacement for Authentication Manager 7.1, 6.1 and 5.2 using the administrative interface:

1. Logon to the Authentication Manager administrative interface:
   - For Authentication Manager 6.1 and 5.2 this is through Quick Admin or the Remote Desktop Client.
   - For Authentication Manager 7.1 this is through the Security Console.
2. Import new seed records into the Authentication Manager Primary server.
3. Look up the specific user whose token you want to replace and identify the current token assigned to that user.
4. Note the assigned RSA SecurID token, and proceed to edit the currently assigned RSA SecurID Token to associate a new replacement RSA SecurID Token. This functionality permits the automated replacement through a user initiated authentication.
5. Repeat for every user that requires a replacement RSA SecurID Token.
6. Validate in the Authentication Manager administrative interface that the operations were successful.
7. Create a report of users with their assigned replacement tokens for distribution purposes.
8. Execute your communication plan to inform users and to set expectations.
9. Distribute tokens to the correct users in a secure manner that is convenient for the organization.
10. Users perform an authentication (login) to an RSA agent using the newly assigned token. The Login is performed with normal passcode (current PIN + tokencode from new token).

11. The original token is replaced by a new (replacement) token.
    - Original token is unassigned from user.
    - Original token may be deleted from database simultaneously.
    - Same PIN is maintained on the replacement token.

    Note: For information about PIN policy best practices, see the *RSA Authentication Manager Security Best Practices Guide*.

12. Run reports on regular intervals to ensure token replacements occur within days of receipt. If token replacements are taking too long, disable the user's account or existing token forcing them to contact an administrator or appropriate help desk personnel.

Detailed steps for token replacement for Authentication Manager 7.1 5.2 and 6.1 can be found in Appendix A and Appendix B respectively.

RSA SecurID Token Replacement Using Bulk Scripting

When replacing large numbers of hardware tokens, RSA recommends that you use bulk administration tools or scripts. This is the high-level process for token replacement for Authentication Manager 7.1, 6.1 and 5.2 using scripts:

1. Import new seed records into the Authentication Manager Primary server.
2. Install the appropriate scripts on the Authentication Manager primary server.
3. Use the reporting capabilities of each platform to identify users and physical tokens that must be replaced at each location.
4. Generate a list of token serial numbers and associated users for which tokens are being replaced in accordance with your token replacement strategy.
5. Generate a list of replacement token serial numbers (attempt to group by box/tray of tokens to facilitate distribution).
6. Create a coma separated values (.csv) input file for the associate scripts in the required format.
7. Run scripts to replace tokens for list of users.

    Note: You should test the scripts and input files in a test environment to make sure they achieve the desired results.

8. Manually validate in the Authentication Manager administrative interface that the operation was successful.
9. Destroy copies of the .csv files and Excel files used to create the replacement loader file.
10. Create a report of users with their assigned replacement tokens for distribution purposes.
11. Execute your communication plan to inform users and to set expectations.

12. Distribute tokens to the correct users in a secure manner that is convenient for the organization.

13. User performs an authentication (login) to an RSA agent using the newly assigned token; the Login is performed with normal passcode (current PIN + tokencode from new token).

14. The original token is replaced by a new (replacement) token.

    - Original token is unassigned from user.
    - Original token may be deleted from database simultaneously.
    - Same PIN is maintained on the replacement token.

    Note: For information about PIN policy best practices, see the *RSA Authentication Manager Security Best Practices Guide*.

15. Run reports on regular intervals to ensure token replacements occur within days of receipt. If token replacements are taking too long, disable the user's account or existing token forcing them to contact an administrator or appropriate help desk personnel.

16. Destroy all copies of the CSV files, Excel files, and user and token lists used for the token replacement activity.

The token replacement scripts will vary slightly depending if the customer's environment is Authentication Manager 7.1 or Authentication Manager 6.1 (and prior). Detailed steps for token replacement on Authentication Manager 7.1, 6.1 and 5.2 using scripts can be found at Appendix C and Appendix D respectively.

For more information about bulk administration tools, contact your Sales Representative.

Testing

The following are essential test considerations:

- Tokens and seed records utilized in the test environment should not be re-used in production after the testing phase.
- Scripts and tools, if utilized, for performing bulk operations should always be tested in a development or test environment which replicates the production environment as much as possible.
- When applying the token replacement solution to the production environment, apply to a smaller group of pilot users before applying to the larger user population.
- Testing should include scenarios for at least the following tasks:
  - End-user token activation procedure(s).
  - If utilized, testing the token import scripts and input files.
  - Validating scripts for generating user and token assignment reports.

Restrictions and Limitations

All guidance and information set forth in this white paper and any other documents, including, without limitation, guidance and other information published in the Best Practices Guides, regarding the creation of a phased and prioritized hardware token replacement strategy is general guidance and is provided “as is” and EMC makes no representations or warranties of any kind with respect to the information in this white paper and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

RSA strongly recommends that each organization’s actual plan be established in accordance with its business requirements, existing policies and procedures, communication plans or mechanisms and any applicable regulatory compliance regimes.

Appendix A: RSA Authentication Manager 7.1 Token Replacement Using Security Console

Overview

This appendix describes the process for replacing hardware tokens in Authentication Manager 7.1. The "Replace SecurID Token" functionality is designed for replacing an existing hardware token with a new token where the end-user's PIN is retained.

To activate a replacement token, the end-user must authenticate with the new token at which point the Authentication Manager server unassigns and disables or deletes the old token from the database and assigns the existing PIN to the new token.

Using the "Replace SecurID Token" option is the recommended method for issuing new tokens to users due to the superior end-user experience while also reducing administrative activity, such as assigning end-users new tokens and manually disabling the old token. Software or hardware tokens can be used to replace existing software or hardware tokens in the system. This guidance is for replacing RSA SecurID hardware tokens with new RSA SecurID hardware tokens only.

There are two separate options when using "Replace SecurID Token":

- replacing a token by manually selecting a specific token from a list of available tokens; or
- replacing a token with the next available token from the pool of available tokens.

The "next available" replacement process is only suitable for customers with a limited number of tokens distributed from a central location since the automatic selection process randomly selects the next unassigned token with the lowest serial number.

For hardware tokens this requires the exact token to be located and retrieved from the correct tray of tokens. It may also have the additional effect of assigning a different RSA SecurID token type, for example, an RSA SecurID Software Token, which may not be the desired result. RSA recommends the manual selection option described below.

Token Replacement Process – Manually Selected Token

The following instructions describe how to assign a replacement token from a searchable list of available, unassigned tokens. It is best suited for assigning replacement hardware "fob" tokens, especially if tokens are issued or shipped from multiple locations, or where software and hardware tokens are deployed.

This process affords the best control over assigning replacements so the administrator can make the appropriate choice based on their business and/or logistical requirements.

To replace an RSA SecurID hardware token with a selected RSA SecurID hardware token:

1. Log on to the RSA Security Console.
2. Lookup the user and then token. Click **Identity** > **Users** > **Manage Existing**.

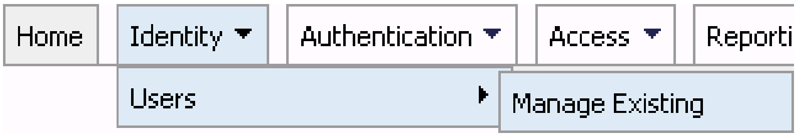
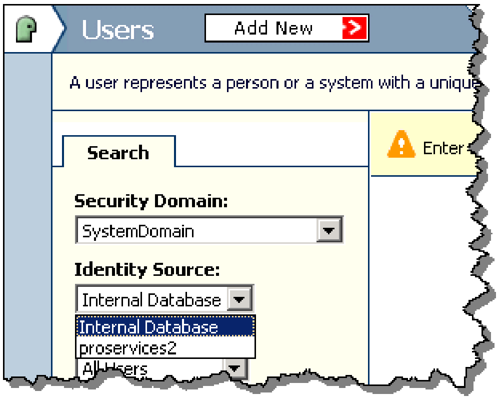


Figure 1- User Lookup

3. Select the correct Identity Source.



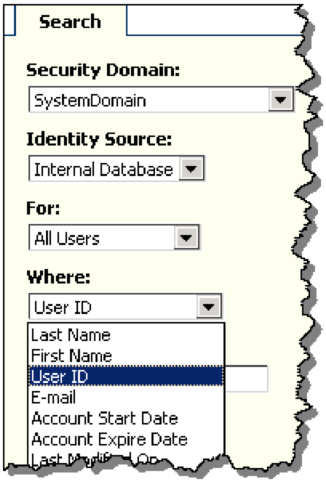
A user represents a person or a system with a unique

Security Domain: SystemDomain

Identity Source: Internal Database

Figure 2- Select Identity Source

4. By default the UI searches on last name. Optionally pick a different user attribute from the list if desired. For example, you can pick User ID.



Security Domain: SystemDomain

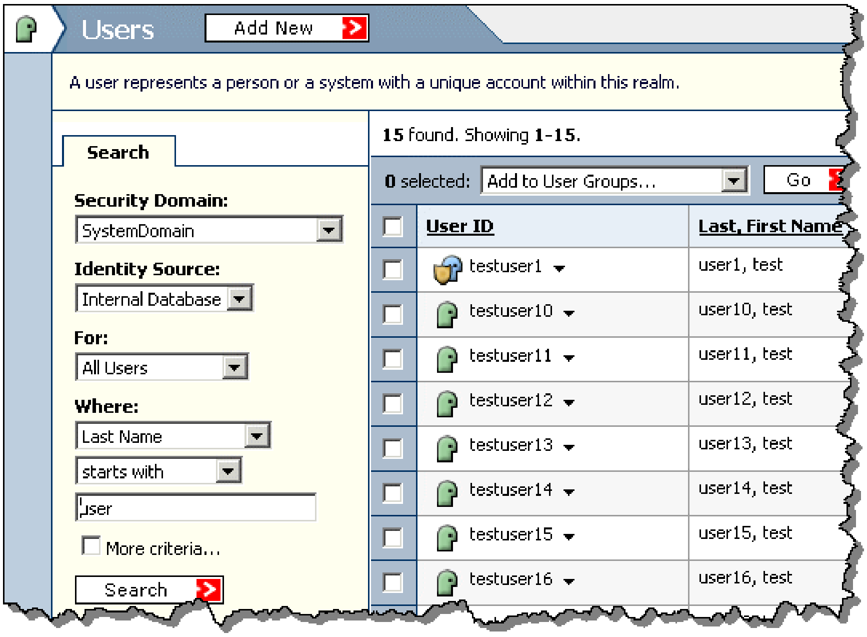
Identity Source: Internal Database

For: All Users

Where: User ID

Figure 3- User Search Attribute Screen

5. Enter identifying user information in the search box and execute the search for the user.



A user represents a person or a system with a unique account within this realm.

Security Domain: SystemDomain

Identity Source: Internal Database

For: All Users

Where: Last Name

starts with

user

☐ More criteria...

15 found. Showing **1-15**.

0 selected: Add to User Groups... Go

☐	User ID	Last, First Name
☐	testuser1 ▼	user1, test
☐	testuser10 ▼	user10, test
☐	testuser11 ▼	user11, test
☐	testuser12 ▼	user12, test
☐	testuser13 ▼	user13, test
☐	testuser14 ▼	user14, test
☐	testuser15 ▼	user15, test
☐	testuser16 ▼	user16, test

Figure 4- User Search

6. Click the ▼ menu icon next to the UserID and select **SecurID Tokens** from the menu.

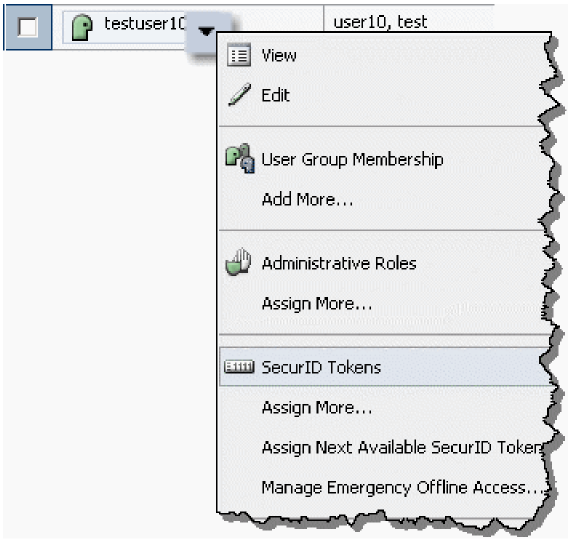
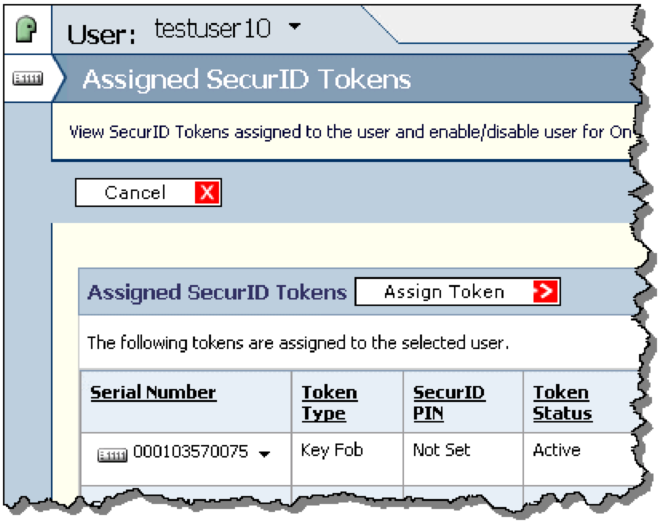


Figure 5- User Context Menu

7. The tokens assigned to the user will be listed.



View SecurID Tokens assigned to the user and enable/disable user for On

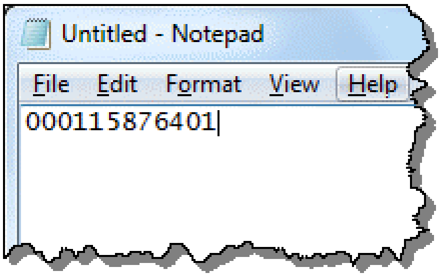
Assigned SecurID Tokens

The following tokens are assigned to the selected user.

Serial Number	Token Type	SecurID PIN	Token Status
000103570075	Key Fob	Not Set	Active

Figure 6- List of tokens assigned to user

8. Make note of the token serial number to be replaced.



000115876401

Figure 7 - Making Note of the token serial number

9. Click **Authentication** > **SecurID Tokens** > **Manage Existing**.

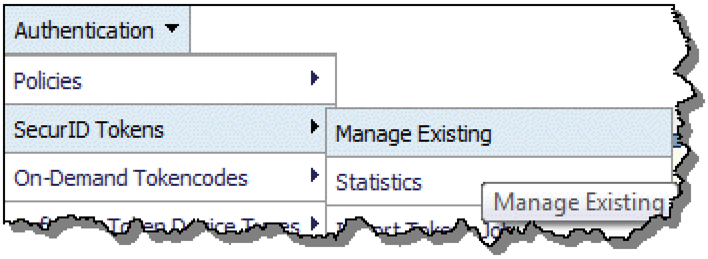
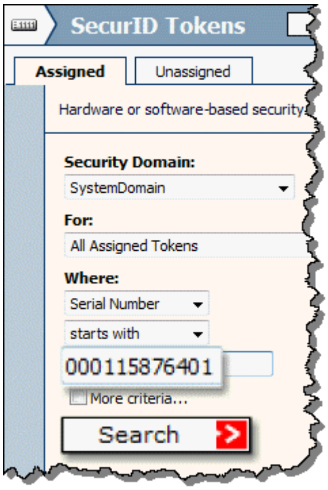


Figure 8- Manage Existing RSA SecurID Tokens

10. Enter the recorded serial number, and if necessary, select the correct security domain from the drop-down list and click **Search**.



Hardware or software-based security

Security Domain:
SystemDomain

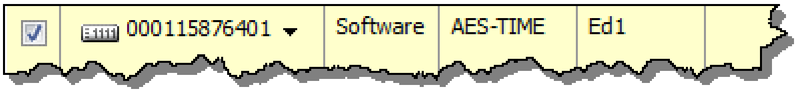
For:
All Assigned Tokens

Where:
Serial Number
starts with
000115876401

□ More criteria...

Figure 9- Search for Chosen Token to be a Replacement

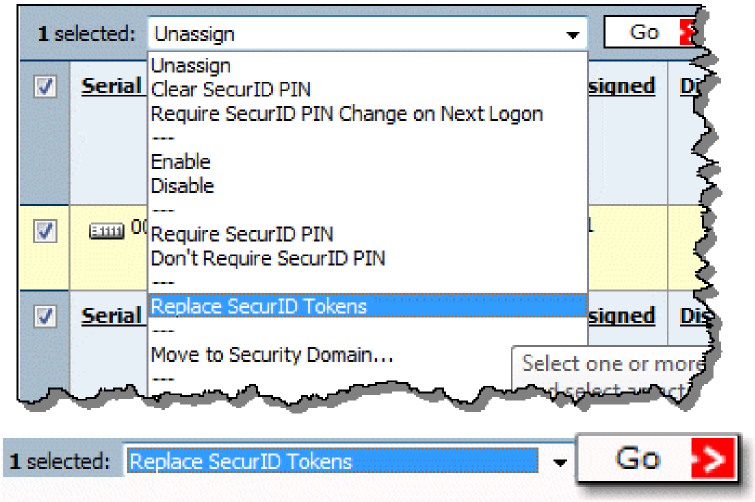
11. Select the token to be replaced



☑	000115876401	Software	AES-TIME	Ed1

Figure 10- Select the Token to Be Replaced

12. From the Action menu, select **Replace SecurID Tokens,** and click **Go**.

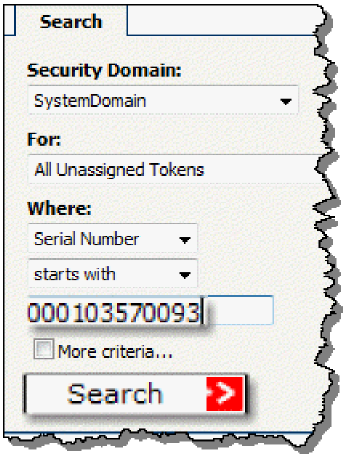


1 selected: Unassign

1 selected: Replace SecurID Tokens

Figure 11- Select Replace SecurID Tokens From Action Menu

13. Select an RSA SecurID hardware token from the tray and enter the serial number in the search dialog.



Search

Security Domain: SystemDomain

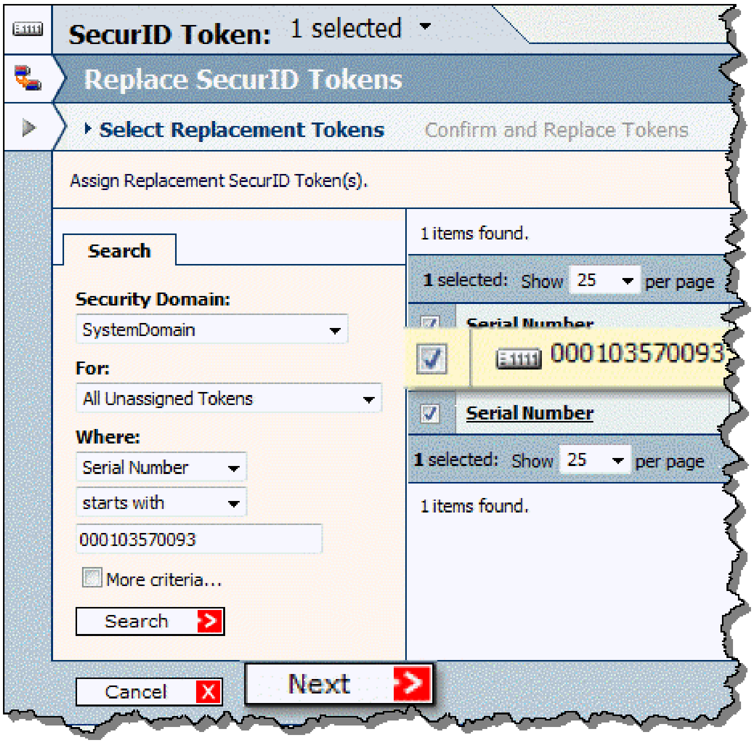
For: All Unassigned Tokens

Where: Serial Number starts with 000103570093

□ More criteria...

Figure 12- Enter Serial Number of New Token

14. Select the token and click **Next**.



SecurID Token: 1 selected

Replace SecurID Tokens

Select Replacement Tokens | Confirm and Replace Tokens

Assign Replacement SecurID Token(s).

Security Domain: SystemDomain

For: All Unassigned Tokens

Where: Serial Number starts with 000103570093

☐ More criteria...

1 items found.

1 selected: Show 25 per page

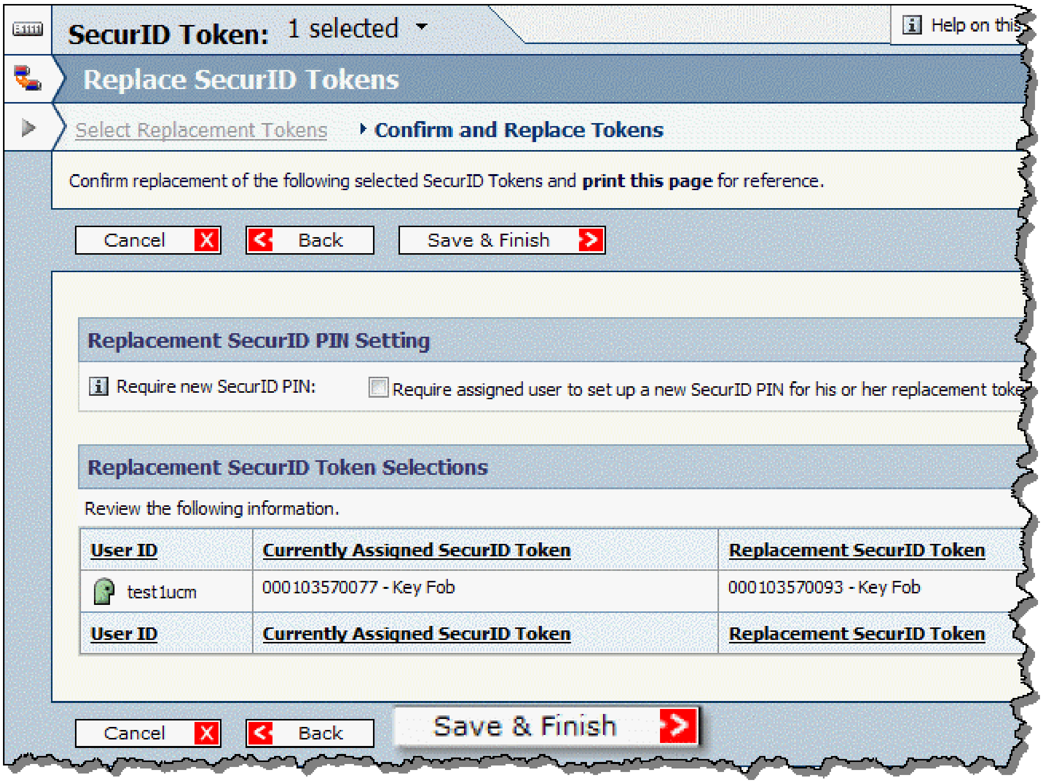
☑	Serial Number
☑	000103570093

1 selected: Show 25 per page

1 items found.

Figure 13- Select the Chosen Replacement Token

15. Click **Save & Finish** to complete the hardware token replacement.


SecurID Token: 1 selected
Help on this
Replace SecurID Tokens
Select Replacement Tokens
Confirm and Replace Tokens
Cancel
Back
Save & Finish

Confirm replacement of the following selected SecurID Tokens and **print this page** for reference.

Replacement SecurID PIN Setting

Require new SecurID PIN: □ Require assigned user to set up a new SecurID PIN for his or her replacement toke

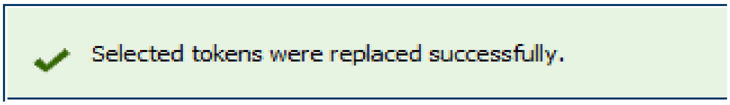
Replacement SecurID Token Selections

Review the following information.

User ID	Currently Assigned SecurID Token	Replacement SecurID Token
test1ucm	000103570077 - Key Fob	000103570093 - Key Fob
User ID	Currently Assigned SecurID Token	Replacement SecurID Token

Figure 14- Confirm the Replacement

16. The selected hardware token has now been activated as a replacement.



✓ Selected tokens were replaced successfully.

Figure 15- Hardware Token as Replacement Complete

This is the end of the hardware token replacement process. You must repeat this process for each token that is being replaced.

Appendix B: RSA Authentication Manager 5.2 and 6.1 Token Replacement Using Administrative Interface

Overview

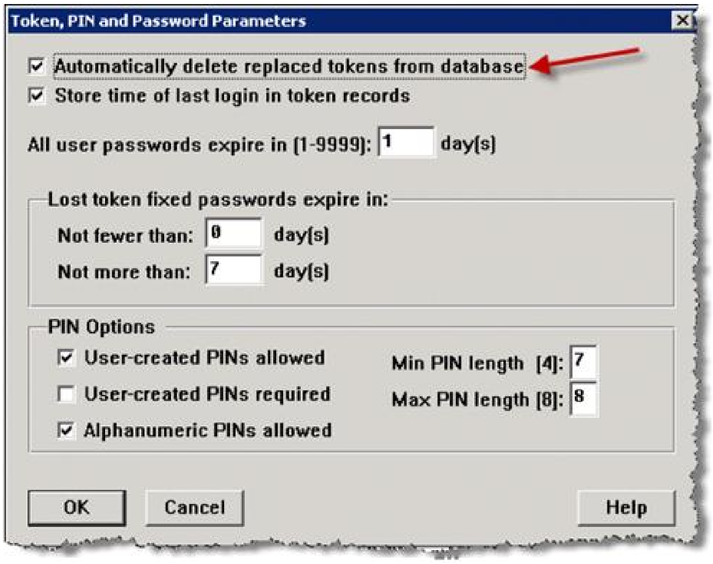
This document describes the manual process for replacing hardware tokens in Authentication Manager 5.2 and6.1. The “Assign Replacement Token” functionality is designed for replacing an existing token with a new token where the user’s PIN is retained. To activate a replacement token, the end-user must authenticate with the new token. After a successful authentication, Authentication Manager unassigns and disables or deletes the old token from the database, and assigns the existing PIN to the new token.

Using the “Assign Replacement Token” option is the recommended method for issuing new tokens to users due to the superior end-user experience while also reducing administrative activity, such as assigning users new tokens and manually disabling the old token. Software or hardware tokens can be used to replace existing software or hardware tokens in the system. This guidance is for replacing hardware tokens with new hardware tokens only.

Token Replacement Process – Manually Selected Token

To replace an RSA SecurID hardware token with a selected RSA SecurID Hardware token:

1. Log on to Authentication Manager through Quick Admin or the Database Administration application in remote mode.
2. Click **System Configuration** > **System Parameters** > **Token, PIN and Password Parameters**. If not already selected, select **Automatically delete replaced tokens from database**. Selecting this option completely removes the replaced token from the database after a successful authentication with the replacement token.



Token, PIN and Password Parameters

☑ Automatically delete replaced tokens from database

☑ Store time of last login in token records

All user passwords expire in (1-9999): 1 day(s)

Lost token fixed passwords expire in:

Not fewer than: 0 day(s)

Not more than: 7 day(s)

PIN Options

☑ User-created PINs allowed

☐ User-created PINs required

☑ Alphanumeric PINs allowed

Min PIN length (4): 7

Max PIN length (8): 8

Figure 16- Enable Automatic Delete of Replaced Token Screen

3. Lookup user and then token. Click **User** > **Edit User**.

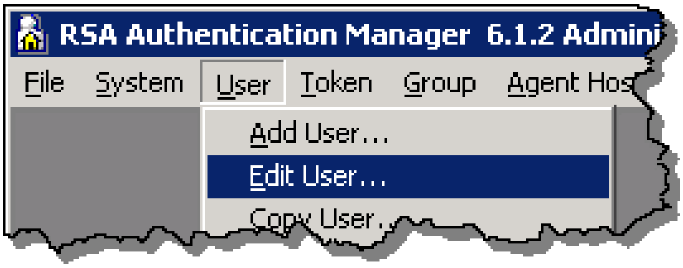
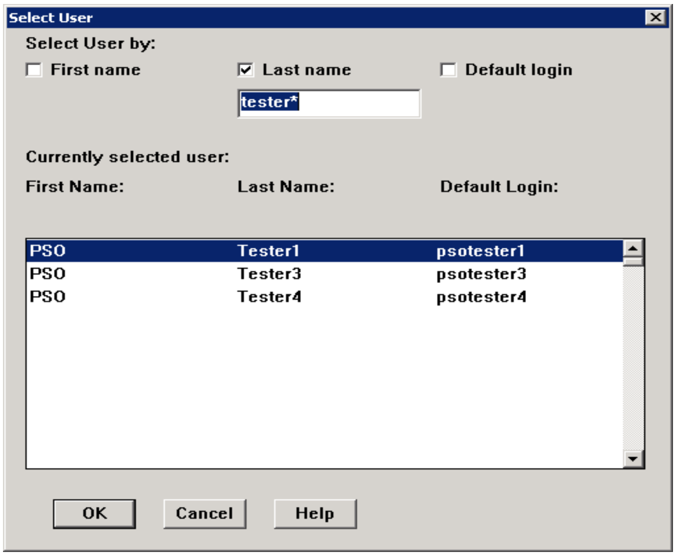


Figure 17- User Lookup

4. Select a user. You can search for a user by First name, Last name or Default Login (UserID).



Select User

Select User by:

☐ First name ☑ Last name ☐ Default login

tester*

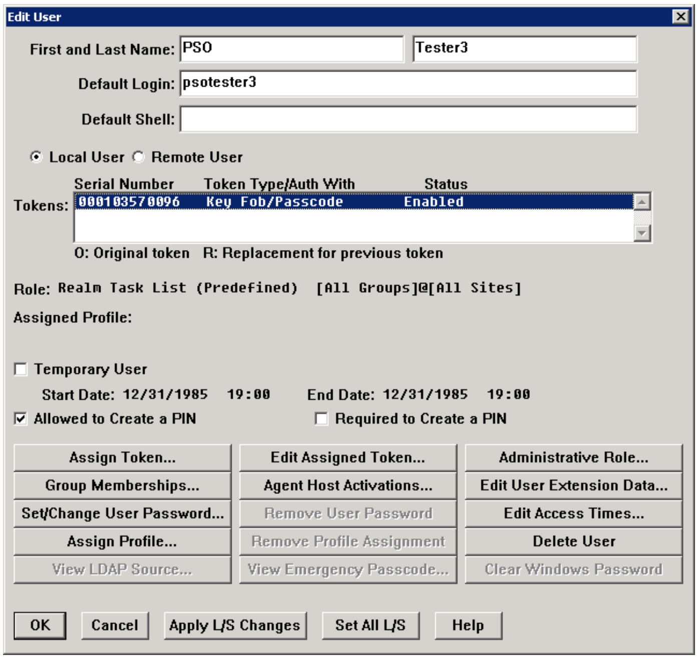
Currently selected user:

First Name: Last Name: Default Login:

PSO	Tester1	psotester1
PSO	Tester3	psotester3
PSO	Tester4	psotester4

Figure 18- User Search Page

5. Once the user is selected, a page displays with the user's information. This page contains information about the user's current token and administrative actions that can be performed against the token.



Edit User

First and Last Name: PSO Tester3

Default Login: psotester3

Default Shell:

● Local User ○ Remote User

	Serial Number	Token Type/Auth With	Status
Tokens:	000103570096	Key Fob/Passcode	Enabled

O: Original token R: Replacement for previous token

Role: Realm Task List (Predefined) [All Groups]@[All Sites]

Assigned Profile:

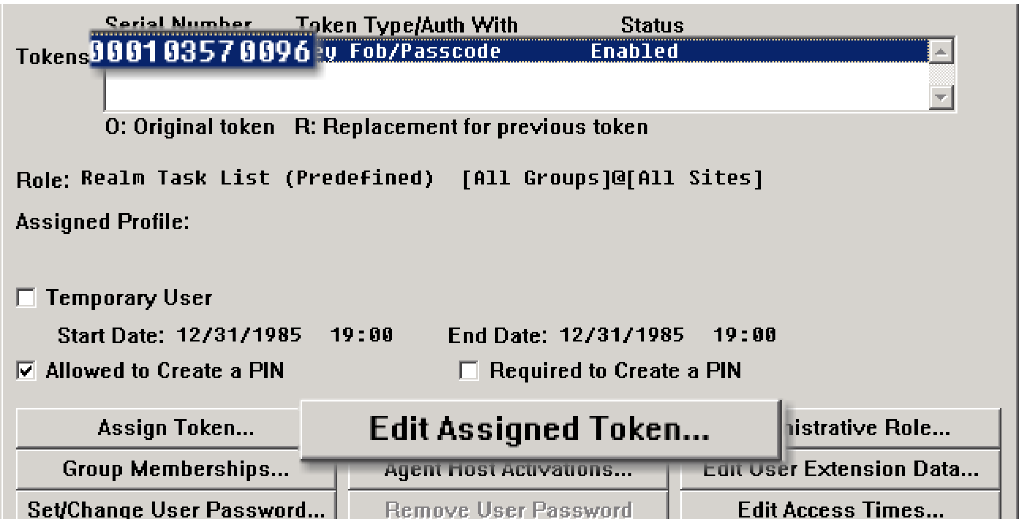
□ Temporary User

Start Date: 12/31/1985 19:00 End Date: 12/31/1985 19:00

☑ Allowed to Create a PIN □ Required to Create a PIN

Figure 19- User Token Information Page

6. Click **Edit Assigned Token**.



	Serial Number	Token Type/Auth With	Status
Tokens	000103570096	ey Fob/Passcode	Enabled

O: Original token R: Replacement for previous token

Role: Realm Task List (Predefined) [All Groups]@[All Sites]

Assigned Profile:

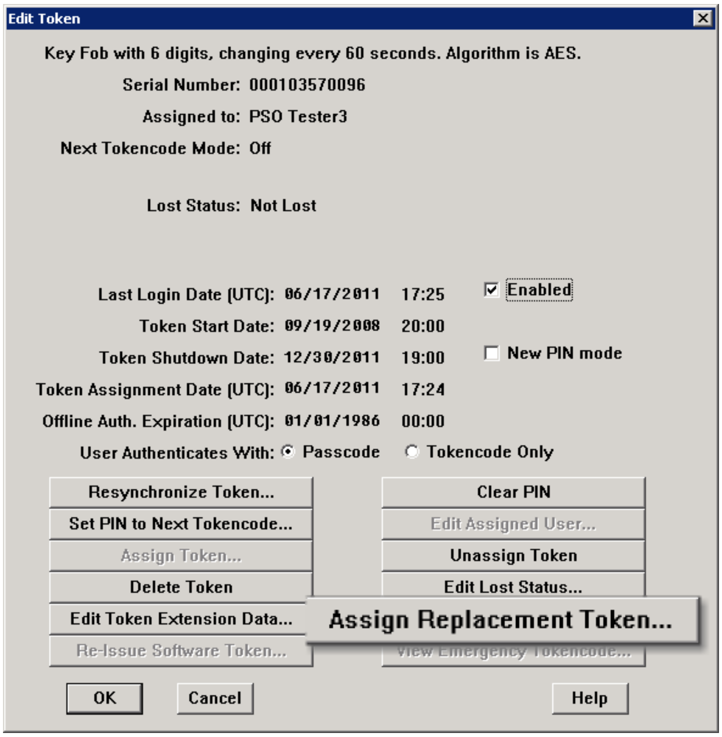
□ Temporary User

Start Date: 12/31/1985 19:00 End Date: 12/31/1985 19:00

☑ Allowed to Create a PIN □ Required to Create a PIN

Figure 20- Edit Assigned Token Option

7. Click **Assign Replacement Token**.



Edit Token

Key Fob with 6 digits, changing every 60 seconds. Algorithm is AES.

Serial Number: 000103570096

Assigned to: PSO Tester3

Next Tokencode Mode: Off

Lost Status: Not Lost

Last Login Date (UTC): 06/17/2011 17:25

Token Start Date: 09/19/2008 20:00

Token Shutdown Date: 12/30/2011 19:00

Token Assignment Date (UTC): 06/17/2011 17:24

Offline Auth. Expiration (UTC): 01/01/1986 00:00

☑ Enabled

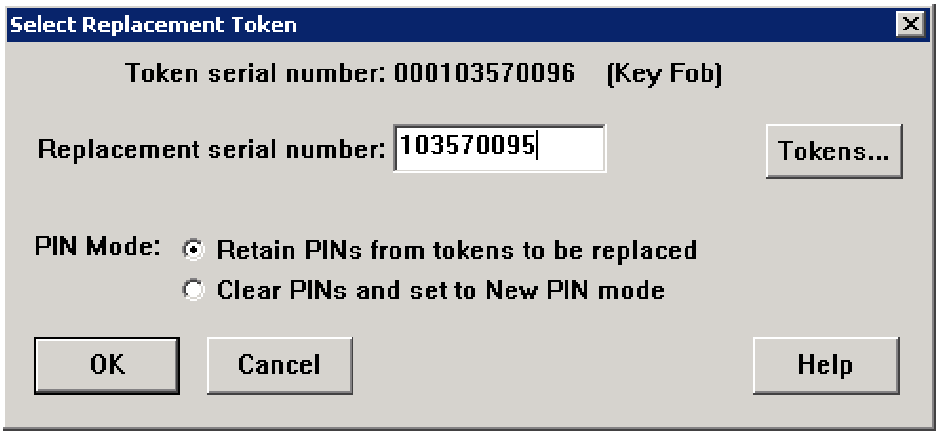
☐ New PIN mode

User Authenticates With: ● Passcode ○ Tokencode Only

Resynchronize Token...
Clear PIN
Set PIN to Next Tokencode...
Edit Assigned User...
Assign Token...
Unassign Token
Delete Token
Edit Lost Status...
Edit Token Extension Data...
Assign Replacement Token...
Re-Issue Software Token...
View Emergency Tokencode...
OK
Cancel
Help

Figure 21- Assign Replacement Token Screen

8. Enter replacement token serial number or click **Tokens** to list available tokens. Ensure **Retain PINs from token to be replaced** is selected.



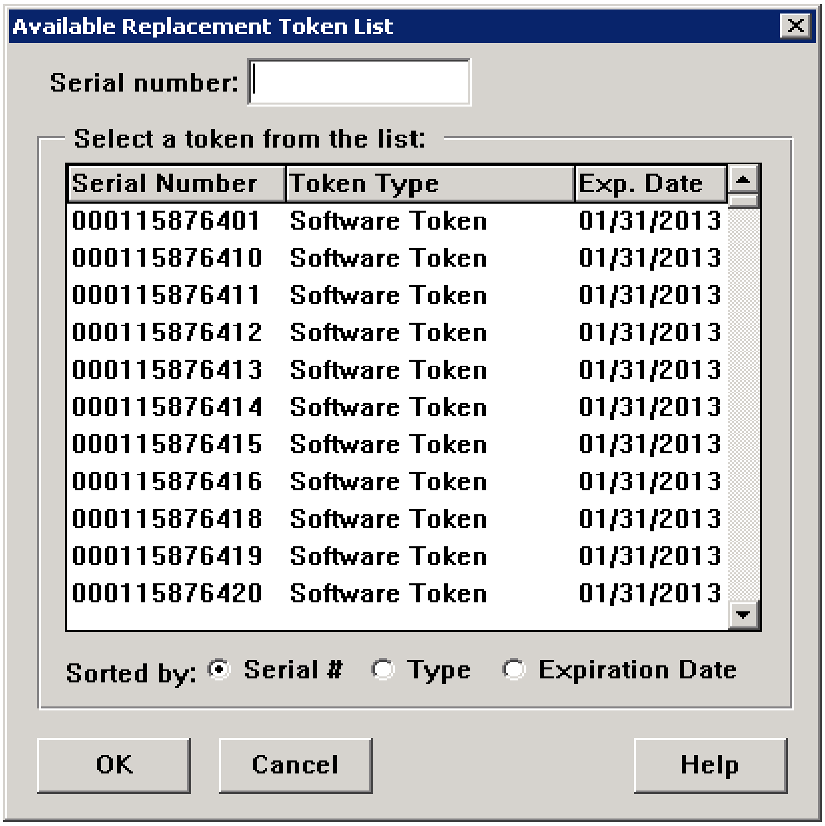
Select Replacement Token

Token serial number: 000103570096 (Key Fob)

Replacement serial number: 103570095

PIN Mode: ● Retain PINs from tokens to be replaced
○ Clear PINs and set to New PIN mode

Figure 22- Entering Token Serial number on ‘Select Replacement Token’ Screen



Available Replacement Token List

Serial number:

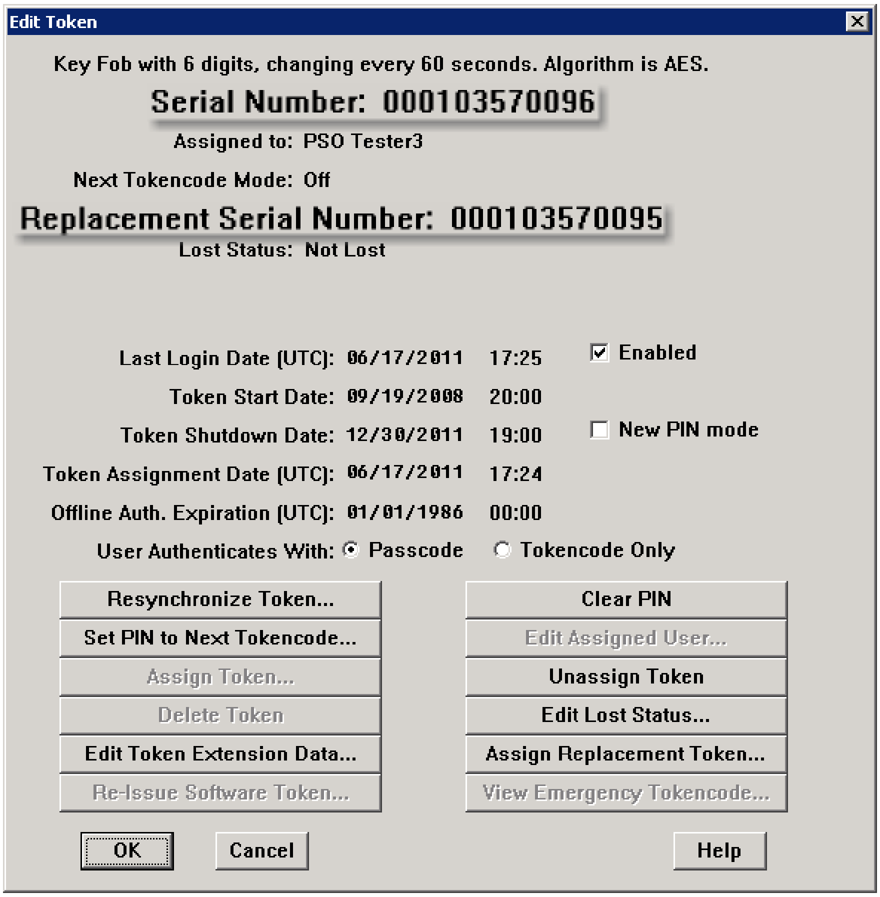
Select a token from the list:

Serial Number	Token Type	Exp. Date
000115876401	Software Token	01/31/2013
000115876410	Software Token	01/31/2013
000115876411	Software Token	01/31/2013
000115876412	Software Token	01/31/2013
000115876413	Software Token	01/31/2013
000115876414	Software Token	01/31/2013
000115876415	Software Token	01/31/2013
000115876416	Software Token	01/31/2013
000115876418	Software Token	01/31/2013
000115876419	Software Token	01/31/2013
000115876420	Software Token	01/31/2013

Sorted by: ● Serial # ○ Type ○ Expiration Date

Figure 23- Listing Available Tokens on 'Select Replacement Token' Screen

9. The screen will display information about the token and additional actions that can be performed against the token. It will also list the replacement token information.



Edit Token

Key Fob with 6 digits, changing every 60 seconds. Algorithm is AES.

Serial Number: 000103570096

Assigned to: PSO Tester3

Next Tokencode Mode: Off

Replacement Serial Number: 000103570095

Lost Status: Not Lost

Last Login Date (UTC): 06/17/2011 17:25

Token Start Date: 09/19/2008 20:00

Token Shutdown Date: 12/30/2011 19:00

Token Assignment Date (UTC): 06/17/2011 17:24

Offline Auth. Expiration (UTC): 01/01/1986 00:00

☑ Enabled

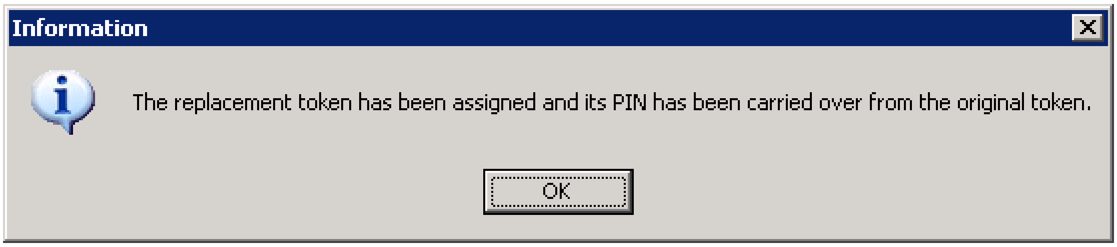
☐ New PIN mode

User Authenticates With: ● Passcode ○ Tokencode Only

Resynchronize Token...
Set PIN to Next Tokencode...
Assign Token...
Delete Token
Edit Token Extension Data...
Re-Issue Software Token...
Clear PIN
Edit Assigned User...
Unassign Token
Edit Lost Status...
Assign Replacement Token...
View Emergency Tokencode...
OK
Cancel
Help

Figure 24- Replacement Token Information Page

10. Click **OK** to accept all changes. The next screen provides a confirmation that the replacement token has been assigned to the user.



Information

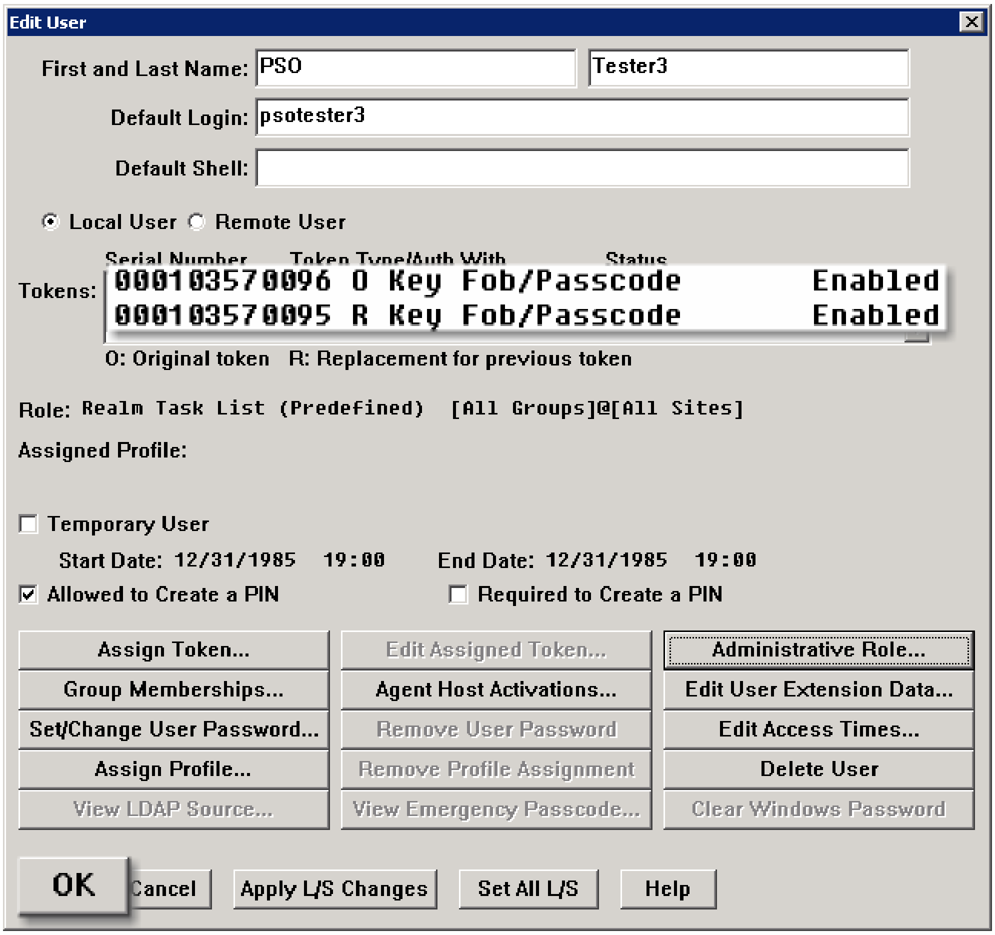
The replacement token has been assigned and its PIN has been carried over from the original token.

Figure 25- Successful Token Replacement Dialog box

Verify Replacement Tokens

To verify the replacement token has been assigned to the user:

1. Lookup the user and token. Click **User** > **Edit User**.
2. In the 'Tokens' section, look for 'R' next to the serial number of the token recently assigned. This verifies that the user now has a replacement token.



Edit User

First and Last Name: PSO Tester3

Default Login: psotester3

Default Shell:

● Local User ○ Remote User

Tokens:

Serial Number		Token Type/Auth With	Status
000103570096	O	Key Fob/Passcode	Enabled
000103570095	R	Key Fob/Passcode	Enabled

O: Original token R: Replacement for previous token

Role: Realm Task List (Predefined) [All Groups]@[All Sites]

Assigned Profile:

□ Temporary User

Start Date: 12/31/1985 19:00 End Date: 12/31/1985 19:00

☑ Allowed to Create a PIN □ Required to Create a PIN

Figure 26- Replacement token verification screen

3. Click **OK**.

Appendix C: RSA Authentication Manager 7.1 Token Replacement Using Scripts

Overview

The Authentication Manager 7.1 Software Development Kit (SDK) can be used to replace or modify large quantities of RSA SecurID information, such as tokens in a customer environment. The Authentication Manager 7.1 SDK supplements administrative features of the Authentication Manager 7.1 Security Console, and enables RSA SecurID administrators to perform bulk administration functions through java utilities from the command line interface or in a background mode through a scheduled script.

For more information on the proper use of Authentication Manager 7.1 SDK, see the *RSA Authentication Manager 7.1 Developer's Guide*.

There is no completely Java-based scripting solution for Authentication Manager 7.1 requiring JAR files to be compiled before they can be executed on the platform, however, support for Jython, a Python /Java derivative, is included and can provide for run-time script execution. Essentially, Jython scripts are hybrid Java applications that are executed without the need for compilation into Java bytecode.

To replace a small number of tokens, RSA recommends using the Authentication Manager 7.1 Security Console to manually replace tokens. To replace a large number of tokens, RSA recommends that you use automation tools to bulk assign replacement tokens. This can be accomplished by a Java developer familiar with the Authentication Manager 7.1 SDK, or by a developer who writes a Jython script. When assigning replacement tokens, RSA recommends that the current PIN be maintained on the replacement token so that the token is not placed in New PIN mode.

Note: For information about PIN policy best practices, see the *RSA Authentication Manager Security Best Practices Guide*.

To facilitate bulk token replacements a Jython script has been included below.

The following is a high level overview of the process:

- Run a script to assign replacement tokens.
- The user performs a login with replacement token as normal.
- Original token is unassigned or removed from database.

Replacing Tokens in Authentication Manager 7.1

Reporting features are provided below to assist with determining User IDs and serial numbers for the tokens to be replaced. As previously mentioned, it is best not to attempt a large number of hardware token replacements at the same time you are changing PIN policies.

Note: For information about PIN policy best practices, see the *RSA Authentication Manager Security Best Practices Guide*.

The token replacement process requires the user to use their current PIN with the new token. Once the new token is activated, the replaced token should be deactivated and deleted

To replace tokens:

1. Before you make any changes, run a full database backup. Store your back up files in a secure location, such as a safe. For details regarding database backups see the *RSA Authentication Manager 7.1 Administrator's Guide*.
2. Build the list of User IDs and serial number targeted for replacement using reports and scripts based on priority, location, and so on. Note that your serial numbers and User IDs are confidential information and should be carefully managed as such.
3. Load the new token seeds into the production Authentication Manager Primary. Remove any trace of the XML file and physically secure the original media in a secure location, such as a safe.
4. Assign serial numbers to the users (serial numbers are sequential) so you have a file containing User ID, old tokenserial#, new tokenserial#. Typically, Microsoft Excel is used to quickly build this file.

   Note: User ID is not required for the script. However, it is helpful to include it for reference.
5. Export the information created above in CSV format with no header row.

   ```
   SAMPLE INPUT FILE:
   10203455,6200113344
   10203456,6200113345
   10203457,6200113346
   ```
6. Copy the CSV file to the Authentication Manager Primary (for example, to the c:\Program Files\RSA Security\RSA Authentication Manager\utils directory) along with the **ReplaceTokensUtils.py** script (see below).
7. Run the script passing in the input file,
   For example, enter
   rsautil jython ReplaceTokensUtil.py -u <admin> -p <password> -f tokens.csv -o result.log; (inputfile.csv being the file created above).

   For more information about managing and executing Jython scripts and customer reports, see the *RSA Authentication Manager 7.1 Developer's Guide*.
8. Use the Authentication Manager Security Console to manually validate that the operation was successful.
9. Destroy copies of the CSV files and Excel files used to create the replacement loader file.
10. Execute the communication plan to inform users and to set expectations.
11. Distribute tokens to users in a secure manner.
12. The user performs an authentication (login) to an RSA Agent using the newly assigned token. The login is performed with normal passcode (current PIN + tokencode from the new token).

13. The original token is replaced by a new (replacement) token.
    - Original token is unassigned from user.
    - Original token can be deleted from database simultaneously.
    - Same PIN is maintained on the replacement token.

Note: For information about PIN policy best practices, see the *RSA Authentication Manager Security Best Practices Guide*.

14. Run reports at regular intervals to ensure token replacements occur within days of receipt. If token replacements are taking too long, disable the users' accounts or existing tokens forcing them to contact an administrator or appropriate help desk personnel.

Jython Scripts

ReplaceTokensUtil.py

Takes list of oldtoken, newtoken as input from stdin.

Error message if replace fails, "old replaced with new" using serial numbers on success.

Example Input File (*inputfile.csv*)

```
10203455,6200113344
10203456,6200113345
10203457,6200113346
```

Java Script Source

**** Note:** The formatting and indentation of the "ReplaceTokensUtil.py" is important when run under Python/jython. Copy and pasting from this document may not provide the desired results and produce runtime errors. The source "ReplaceTokensUtil.py" files can be obtained from SecurCare online at: https://knowledge.rsasecurity.com/docs/rsa_securid/rsa_auth_mgr/71/ReplaceTokensUtils.py

ReplaceTokensUtil.py

```
# Replace users tokens in bulk using a simple CSV file format
# with lines on the format:
# oldserialno,replacementserialno
#
# The numbers must be 6-12 digits with whitespace allowed in the file.
#
# @author avolanis
# @author scheip

import sys

from java.io import BufferedReader
from java.io import FileReader
from java.io import BufferedWriter
from java.io import FileWriter

from java.lang import String, Boolean
from java.util import Date

from jargs.gnu import CmdLineParser
```

ReplaceTokensUtil.py

```
from org.apache.commons.lang import StringUtils

from com.rsa.command import ClientSession
from com.rsa.command import Connection
from com.rsa.command import ConnectionFactory
from com.rsa.command import CommandTargetPolicy
from com.rsa.command import CommandException

from com.rsa.admin import SearchSecurityDomainCommand
from com.rsa.admin.data import SecurityDomainDTO

from com.rsa.common.search import Filter

from com.rsa.authmgr.admin.tokenmgt import LookupTokenCommand
from com.rsa.authmgr.admin.tokenmgt import ReplaceTokensCommand
from com.rsa.authmgr.admin.tokenmgt import EnableTokensCommand

from com.rsa.authmgr.admin.tokenmgt.data import ReplaceTokenDTO
from com.rsa.authmgr.admin.tokenmgt.data import TokenDTO

SECURITY_DOMAIN = "SystemDomain"

class TokenProcessError(Exception):
    def __init__(self, msg):
        self.msg = msg
    def __str__(self):
        return repr(self.msg)

class ReplaceTokensUtil:
    def __init__(self):
        self.inFileOpen = 0
        self.outFileOpen = 0
        self.session = None
        self.count = 0

    def replaceTokens(self, args):
        parser = CmdLineParser()

        admUserOpt = parser.addStringOption('u', "user")
        admPwdOpt = parser.addStringOption('p', "password")

        securityDomainOpt = parser.addStringOption('s', "securitydomain")

        fileInOpt = parser.addStringOption('f', "filein")
        fileOutOpt = parser.addStringOption('o', "fileout")

        verboseOpt = parser.addBooleanOption('v', "verbose")
        helpOpt = parser.addBooleanOption('h', "help")

        parser.parse(args)

        admUser = parser.getOptionValue(admUserOpt)
        admPwd = parser.getOptionValue(admPwdOpt)
        self.verbose = parser.getOptionValue(verboseOpt, Boolean.FALSE)

        securityDomain = parser.getOptionValue(securityDomainOpt)

        self.filein = parser.getOptionValue(fileInOpt)
        self.fileout = parser.getOptionValue(fileOutOpt)

        help = parser.getOptionValue(helpOpt, Boolean.FALSE)
```

ReplaceTokensUtil.py

```
        if (help or len(args) < 2):
            self.displayUsage()
            return

        if (StringUtils.isBlank(admUser) or StringUtils.isBlank(admPwd)):
            raise TokenProcessError("user or password was not provided")

        if StringUtils.isBlank(securityDomain):
            securityDomain = SECURITY_DOMAIN

        if StringUtils.isBlank(self.filein):
            raise TokenProcessError("An input file name is required.")

        if StringUtils.isBlank(self.fileout):
            raise TokenProcessError("An output file name is required.")

        self.session = ConnectionFactory.connect(admUser, admPwd)
        CommandTargetPolicy.setDefaultCommandTarget(self.session)

        try:
            print "Starting replace tokens job ******************************"
            securitDomainGUID = self.getSecurityDomainGUIDFromName(securityDomain)

            nextLine = self.readNext()

            while nextLine != None:

                try:
                    tokens = self.parseLine(nextLine)

                    # is current token assigned?
                    # is current token part of a replacement pair?
                    # API calls here
                    # if yes to either question, write error and continue

                    # is replacement token assigned?
                    # is replacemant token expired?
                    # API calls here
                    # if yes to either question, write error and continue

                    # setup the replacement object

                    # log result (good or bad) and continue
                    self.replaceToken(tokens[0], tokens[1], securitDomainGUID)
                    self.writeLine("Token " + tokens[0] + " is replaced by " + tokens[1])
                    self.count = self.count + 1

                except TokenProcessError, e:
                    self.writeLine(e.msg)

                nextLine = self.readNext()

        finally:
            self.closeOutFile()
            if self.session != None:
                self.session.logout()
            print "Replaced " + repr(self.count) + " tokens"
            print "EOJ - replace tokens ******************************"

    def replaceToken(self, originalTokSerial, replacementTokSerial, securityDomainGUID):
```

ReplaceTokensUtil.py

```
        # Force new token to new pin mode
        # Force new token to enabled

        try:
            tokenDTO = self.getTokenDTO(originalTokSerial)

            if StringUtils.isBlank(tokenDTO.getAssignedUserId()):
                raise TokenProcessError("Token: " + originalTokSerial + " is not 
assigned.")

            repTokenDTO = self.getTokenDTO(replacementTokSerial)

            if not StringUtils.isBlank(repTokenDTO.getAssignedUserId()):
                raise TokenProcessError("Replacement Token: " + replacementTokSerial + " 
already assigned.")

            now = Date()
            if repTokenDTO.getDateTokenShutdown().before(now):
                raise TokenProcessError("Replacement Token: " + replacementTokSerial + " is 
expired.")

            tokenGUID = tokenDTO.getId()
            repTokenGUID = repTokenDTO.getId()

            # Setup the replacemant relationship
            repTknDTO = ReplaceTokenDTO()
            repTknDTO.setReplacingTokenGuid(tokenGUID)
            repTknDTO.setReplacementTokenGuid(repTokenGUID)

            replTokensCmd = ReplaceTokensCommand()
            replTokensCmd.setSetNewPin(1)
            replTokensCmd.setRepTknDTO( [repTknDTO] )
            replTokensCmd.execute()

            enabletokcmd = EnableTokensCommand()
            enabletokcmd.setTokenGuids( [repTokenGUID] )
            enabletokcmd.setEnable(1)
            enabletokcmd.execute()

        except CommandException, e:
            raise TokenProcessError("ReplaceTokensUtil.replaceTokens(...) Error: " + 
e.toString())

    def writeLine(self, lineOut):
        if (not self.outFileOpen):
            self.out = BufferedWriter(FileWriter(self.fileout))
            self.outFileOpen = 1

            self.out.write("File open - replace_tokens ******************************")
            self.out.newLine()

        self.out.write(lineOut)
        self.out.newLine()

    def closeOutFile(self):
        if (self.outFileOpen):
            self.writeLine("File close - replace_tokens ******************************")
            self.out.close()
            self.outFileOpen = 0
```

ReplaceTokensUtil.py

```
    def readNext(self):
        try:
            if (not self.inFileOpen):
                self.inFile = BufferedReader(FileReader(self.filein))
                self.inFileOpen = 1

            return self.inFile.readLine()

        except:
            try:
                writeLine("readNext() Error: " + repr(sys.exc_info()[1]))
            except:
                # ignore this error
                return None
            return None

    def parseLine(self, line):
        """
        # line format:
        # <token serial #>,<token serial #>
        #
        # token serial # can be between 4 (arbritary) and 12 characters in length
        # token serial # must consist of all numeric characters (1234567890)
        # leading zeros are not required
        # token serial # can have leading and trailing spaces
        """

        tokens = line.split(",")

        if len(tokens) != 2:
            raise TokenProcessError("Format error, line " + line + " does not contain two
token serial numbers")

        origToken = self.validateSerial("Original", tokens[0])
        replToken = self.validateSerial("Replacement", tokens[1])

        return [ origToken, replToken ]

    def validateSerial(self, pos, serial):
        serial = String(serial).trim()

        if not String(serial).matches("\\d+"):
            raise TokenProcessError(pos + " serial number " + serial + " is not numeric.")

        length = len(serial)
        if length < 4:
            raise  TokenProcessError(pos + " serial number " + serial + " is less than 4
digits.")

        if length > 12:
            raise TokenProcessError(pos + " serial number " + serial + " is greater than 12
digits.")

        # no zero padding needed?
        if length == 12:
            return serial

        # zero pad the serial to be 12 digits
        return String("000000000000").substring(0, 12 - length) + serial

    def getSecurityDomainGUIDFromName(self, securityDomainName):
```

ReplaceTokensUtil.py

```
        sdCmd = SearchSecurityDomainCommand()
        filter = Filter.equal("name", securityDomainName)
        sdCmd.setFilter(filter)
        sdCmd.execute()

        for dto in sdCmd.getSecurityDomains():
            if String(dto.getName()).equalsIgnoreCase(securityDomainName):
                return dto.getGuid()

        raise TokenProcessError("Identity Source or Security Domain provided is invalid or
is not a valid mapping")

    def getTokenDTO(self, serialNumber):
        cmd = LookupTokenCommand()
        cmd.setSerialNumber(serialNumber)
        cmd.execute()
        return cmd.getToken()

    def lookupToken(self, tokenGUID):
        cmd = LookupTokenCommand()
        cmd.setGuid(tokenGUID)
        cmd.execute()
        return cmd.getToken()

    def displayUsage(self):
        print "Usage:"
        print " -u|--user <administrator username>"
        print " -p|--password <administrator password>"
        print " -s|--securitydomain <Security Domain> (defaults to SystemDomain)"
        print " -f|--filein <Input file path and filename>"
        print " -o|--fileout <Output file path and filename>"
        print " -h|--help (Usage report)"
        print " -v|--verbose"

# Main line processing
args = sys.argv[1:]

try:
    util = ReplaceTokensUtil()
    util.replaceTokens(args)
except TokenProcessError, e:
    print "Replace Token Error: " + e.msg
```

Appendix D: RSA Authentication Manager 5.2 and 6.1 Token Replacement Using Scripts

Overview

TCL is a scripting language that can be used in RSA Authentication Manager 5.2 and 6.1 to replace or modify large quantities of RSA SecurID information, such as tokens in a customer environment. TCL Scripting supplements administrative features of the Authentication Manager 5.2 and 6.1 product line, and enables Authentication Manager administrators to perform bulk administration functions from the command line interface or in a background mode through a scheduled script.

For more information on the proper use of TCL scripts and custom reports, see the *RSA Authentication Manager Administration Toolkit Reference Guide* available on RSA SecureCare Online.

To replace a small number of tokens, it may best to use the Authentication Manager 6.1 Remote Administrative Client to replace tokens. To replace a large number of tokens, RSA recommends that you use tools such as the TCL utility to bulk assign replacement tokens. When assigning replacement tokens, RSA recommends that the current PIN be maintained on the replacement token so that the token is not placed in New PIN mode.

The following is a high level overview of the process:

- Run a script to assign replacement tokens.
- The user performs a login with replacement token as normal.
- Original token is unassigned or removed from database.

Note: For information about PIN policy best practices, see the *RSA Authentication Manager Security Best Practices Guide*. .

Replacing Tokens in Authentication Manager 5.2 and 6.1

TCL scripts and report details will be provided below to assist with determining users and serial numbers for the tokens to be replaced. As previously mentioned, it is best not to attempt a large number of hardware token replacements at the same time you are changing PIN policies.

Note: For information about PIN policy best practices, see the *RSA Authentication Manager Security Best Practices Guide*.

The token replacement process will require the user to use their current PIN with the new token.

Once the new token is activated, the replaced token should be deactivated and deleted. To ensure this happens, make sure **Automatically delete replaced tokens from database** is selected on the Token, PIN and Password Parameters screen. By default, this setting is not enabled.

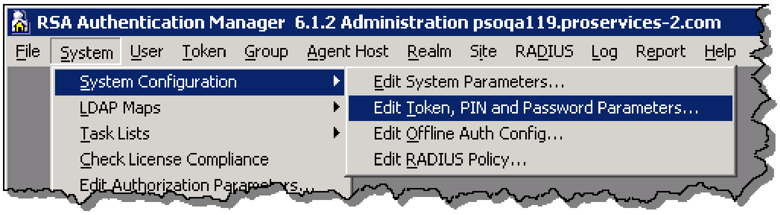
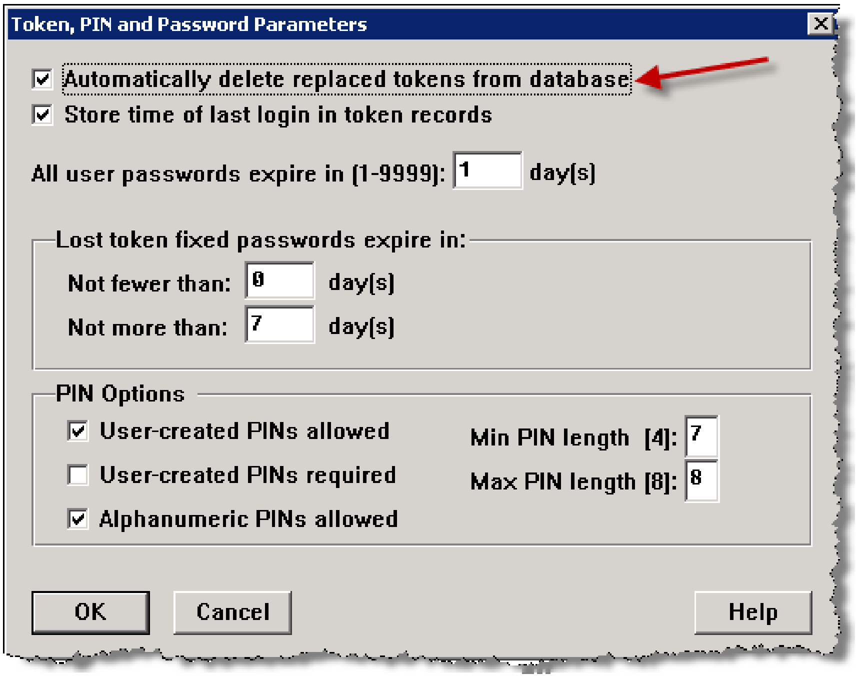

RSA Authentication Manager 6.1.2 Administration psoqa119.proservices-2.com
File System User Token Group Agent Host Realm Site RADIUS Log Report Help
System Configuration
LDAP Maps
Task Lists
Check License Compliance
Edit Authorization Parameters...
Edit System Parameters...
Edit Token, PIN and Password Parameters...
Edit Offline Auth Config...
Edit RADIUS Policy...

Figure 27



☑ Automatically delete replaced tokens from database

☑ Store time of last login in token records

All user passwords expire in (1-9999): 1 day(s)

Lost token fixed passwords expire in:

Not fewer than: 0 day(s)

Not more than: 7 day(s)

PIN Options

☑ User-created PINs allowed

□ User-created PINs required

☑ Alphanumeric PINs allowed

Min PIN length (4): 7

Max PIN length (8): 8

To replace tokens:

1. Before you make any changes, run a full database backup. Store your back up files in a secure location, such as a safe. For details regarding database backups see the RSA Authentication Manager 6.1 Administrator's Guide. For information about securely storing backup files, see the *RSA Authentication Manager Security Best Practices Guide.*

2. Build the list of User IDs and serial number targeted for replacement using reports and scripts based on priority, location, and so on. Note that your serial numbers and User IDs are confidential information and should be carefully managed as such.

3. Load the new token seeds into the production Authentication Manager Primary. Remove any trace of the XML file and physically secure the original media in a secure location, such as a safe.

4. Assign serial numbers to the users (serial numbers are sequential) so you have a file containing userID, old tokenserial#, new tokenserial#. You can use Microsoft Excel to build this file.

Note: UserID is not required for the script. However, it is helpful to include it for reference.

5. Export the created information from 4 above in CSV format with no header row and with just pairs of new and old serial numbers, for example, 000111383838,000112948484.

6. Copy the CSV file to the Authentication Manager Primary (for example, c:\Program Files\RSA Security\RSA Authentication Manager\utils directory) along with **replace-tokens.tcl** script (see below).

7. Run the script passing in the input file,
   For example, enter
   ./utils/tcl/BIN/tcl-sd.exe replace-tokens.tcl < inputfile.csv; (inputfile.csv being the file created from 5 above).

   For more information about managing and executing TCL scripts and customer reports, see the RSA Authentication Manager Administration Toolkit Reference Guide.

8. Validate manually in the Authentication Manager Administrative interface that the operation was successful.

9. Destroy copies of the CSV files and Excels file used to create the replacement loader file.

10. Execute the communication plan to inform users and to set expectations.

11. Distribute tokens to users in a secure manner .

12. The user performs an authentication (login) to an RSA Agent using the newly assigned token. The login is performed with normal passcode (current PIN + tokencode from the new token).

13. The original token is replaced by a new (replacement) token.

    - Original token is unassigned from user.
    - Original token can be deleted from database simultaneously.
    - Same PIN is maintained on the replacement token.

Note: For information about PIN policy best practices, see the *RSA Authentication Manager Security Best Practices Guide*.

14. Run reports on regular intervals to ensure token replacements occur within days of receipt. If token replacements are taking too long, disable the user’s account or existing token forcing them to contact an administrator or appropriate help desk personnel.

TCL Scripts

Replace-Token.tcl

Takes list of oldtoken,newtoken as input from stdin.

Error message if replace fails, “old replaced with new” using serial numbers on success.

Example Input File (*inputfile.csv*)

```
10203455,6200113344
10203456,6200113345
10203457,6200113346
```

TCL Script Contents

replace-token.tcl

```
# replace-tokens.tcl
# read a list of "oldtoken,newtoken" from stdin, one pair per line,
# and make best effort to assign newtoken as a replacement for oldtoken.
#

# fix up some padding for the token serial numbers
set pad(0) 000000000000
set pad(1) 00000000000
set pad(2) 0000000000
set pad(3) 000000000
set pad(4) 00000000
set pad(5) 0000000
set pad(6) 000000
set pad(7) 00000
set pad(8) 0000
set pad(9) 000
set pad(10) 00
set pad(11) 0
set pad(12) ""

# deal with securid init stuff first
Sd_ApiInit "" "" 1

while { [gets stdin lineIn] >= 0 } {

        #puts "RAW $lineIn"
        # pair is in lineIn, maybe -- remove whitespace and see if anything is left
        set inp [split $lineIn ","]
        set oldtoken [string trim [lindex $inp 0] " \""]
        set newtoken [string trim [lindex $inp 1] " \""]
        #puts "IN $oldtoken $newtoken"

        # make sure oldtoken is padded to 12 chars
        set len [string length $oldtoken]
        if { $len < 12 } {
                set foo ""
                set oldtoken [append foo $pad($len) $oldtoken]
        }
        # make sure newtoken is padded to 12 chars
```

replace-token.tcl

```
        set len [string length $newtoken]
        if { $len < 12 } {
                set foo ""
                set newtoken [append foo $pad($len) $newtoken]
        }

        if { [catch {Sd_ReplaceToken $oldtoken $newtoken} errMsg] } {
                puts "ERROR $errMsg ($oldtoken $newtoken)"
        } else {
                puts "$oldtoken replaced with $newtoken"
        }

}
Sd_ApiEnd
```

Customer Support Information

For information, contact RSA Customer Support:

U.S.: 1-800-782-4362, Option #5 for RSA, Option #1 for SecurCare note

Canada: 1-800-543-4782, Option #5 for RSA, Option #1 for SecurCare note

International: +1-508-497-7901, Option #5 for RSA, Option #1 for SecurCare note