

# RSA SecurID Ready Partner Guide

Enabling Partner Products for  
RSA SecurID Two Factor Authentication

Last Updated: April 14, 2006



# Table of Contents

<b>Important Information for RSA Secured Partners .....</b>	<b>4</b>
RSA Security Product Re-Branding .....	4
Document Usage .....	4
Obtaining the RSA Authentication Manager API's .....	4
<b>RSA SecurID Authentication Overview .....</b>	<b>5</b>
RSA SecurID Authentication .....	5
Becoming an RSA Secured Partner .....	5
RSA SecurID Authenticators .....	6
User Passwords .....	6
Two-Factor User Authentication .....	6
RSA SecurID Tokencode Generation and Synchronization .....	7
RSA Security's proprietary algorithm .....	7
Accountability and Security Auditing .....	7
Protection from Intruders .....	8
Detecting a Replay Attack .....	8
<b>Getting Started .....</b>	<b>9</b>
Points of Integration .....	9
Planning the Partner Integration .....	9
RSA Authentication Agent Overview .....	9
Planning RSA SecurID Integration .....	10
Obtaining the Latest RSA SecurID Authentication Agent API's .....	10
<b>RSA Authentication Agent Overview .....</b>	<b>11</b>
RSA Authentication Agent Functionality and Messages .....	11
The "Enter Passcode": Prompt .....	12
RSA Authentication Manager Replica(s) .....	12
Passcode Accepted .....	12
Access denied .....	12
New PIN Mode .....	13
System Generated PIN .....	13
User-Selectable PIN .....	14
User-Defined PIN .....	14
PIN Rejected .....	15
Re-authentication .....	15
Next Tokencode Mode .....	15
SecurID Integration for RADIUS Server Products .....	15
SecurID with Web Based Applications .....	15
How to incorporate Web Access Authentication .....	16
Leveraging the RSA Authentication Agent .....	16
Building in the API's .....	16
Distributing Web Access Authentication Cookies .....	16
Setting Cookie Expiration Times .....	17
Enabling Domain Cookies .....	17
Using the Web Access Authentication HTML Forms .....	17
Customizing Message Strings .....	17
<b>RSA SecurID Authentication via RADIUS Protocol .....</b>	<b>19</b>
<b>The RSA Authentication Manager Admin API .....</b>	<b>20</b>
Remote Management Capabilities .....	20

<b>The RSA SecurID Software Token API .....</b>	<b>21</b>
<b>Documenting the Solution .....</b>	<b>22</b>
How do I fill out the RSA SecurID Implementation Guide? .....	22
Partner Information .....	22
Solution Summary .....	22
Partner Integration Overview .....	23
Product Requirements .....	24
RSA Authentication Manager Configuration .....	25
Partner Authentication Agent Configuration .....	25
Certification Environment .....	26
Certification Checklist .....	26
Known Issues .....	28
Appendix .....	28
<b>Completing RSA Secured Certification Testing .....</b>	<b>29</b>
The Environment .....	29
Problems, Questions and Failures .....	29
Using the RSA Authentication Log Monitor .....	29
Initial Configuration .....	30
Create Test User Entries .....	30
Create Agent Host Entries .....	31
System Generated PIN Configuration (RADIUS ONLY) .....	32
Mandatory Functionality Testing .....	33
Additional Functionality Testing .....	35
Software Token API Testing .....	36
Domain Credential Functionality Testing .....	37

# Important Information for RSA Secured Partners

---

## RSA Security Product Re-Branding

In September, 2004 the RSA ACE/Server and RSA ACE/Agent names were changed to RSA Authentication Manager and RSA Authentication Agent respectively. Throughout this guide, you'll see references to documents that haven't been updated to reflect the name changes.

## Document Usage

This guide is intended for use by registered partners or internal RSA personnel only.

Customers utilizing this guide can obtain assistance for items covered in this document via RSA Security Developer Support. For documentation and pricing information regarding this level of product support, please refer to the following URL:

<http://www.rsasecurity.com/node.asp?id=1265>

## Obtaining the RSA Authentication Manager API's

The RSA SecurID Authentication API's can be downloaded from RSA SecurCare Online:

<https://knowledge.rsasecurity.com>

# RSA SecurID Authentication Overview

---

This guide is to be used for implementing RSA SecurID two-factor authentication or integrating with RSA Authentication Manager and/or RSA Authentication Agents as part of the RSA Secured Partner Program.

## RSA SecurID Authentication

The RSA Authentication Manager® provides authentication services to control access to network resources, routers, applications, and operating systems. The RSA Authentication Manager authenticates the identity of users based on two factors: the current time-based code from the user's assigned RSA SecurID® authenticator and a secret, memorized personal identification number (PIN). This patented technology is part of a system that ensures that access is granted only to authorize users on valid clients of the RSA Authentication Manager.

The system consists of the RSA Authentication Manager software running in a trusted environment and RSA Authentication Agent software on devices and applications that are to be protected by RSA SecurID authentication. Optional replica RSA Authentication Managers provides backup authentication services if the primary server is not running or if the network connection between a client and the master is broken temporarily.

The RSA Authentication Manager integrates a commercial database application developed by Progress Software Corporation to allow for scalability to large numbers of users and authenticators, and to provide programming interfaces for writing custom reports that can include RSA Authentication Manager and other data.

RSA Authentication Manager also includes a toolkit for creating custom administration applications. The Administration Toolkit consists of functions and executables that can read from and write to the RSA Authentication Manager databases

## Becoming an RSA Secured Partner

RSA Security encourages and supports the efforts of developers to add RSA Authentication Agent functionality to their own products, such as remote access servers, firewalls, routers, and software applications.

Under a licensing agreement, developers are provided object or source code that executes in a client system. Its primary function is to handle the interaction between the client and the RSA Authentication Manager software where the implementer is responsible for gathering the authentication data from the user. The Authentication Agent code was designed to require minimal operating system-level support.

RSA Secured Partners are required to complete specific technical and marketing related requirements. These requirements are described in detail in the RSA Secured Partner Program description sent to your organization by RSA Security. If you have not done so already, review the requirements covered in this document and contact RSA Security Partner Development group if you have any questions.

## RSA SecurID Authenticators

Most RSA SecurID authenticators are small, handheld devices containing micro-processors that calculate and display unpredictable codes. These codes change at a specified interval, typically 60 seconds.

Every authorized user on a protected system can be assigned up to three RSA SecurID authenticators to use when accessing a protected resource. The code displayed by an authenticator at the moment the user attempts access is one part of the user's RSA SecurID Passcode, which is required for positive user authentication and system access.

Allowing users up to three authenticators is convenient for both RSA Authentication Manager users and administrators. For example, employees might use different authenticator types from different locations: RSA SecurID authenticating for connecting from home and RSA Software Token authenticators for connecting at the office. Furthermore, administrators can manage large numbers of expiring authenticators by assigning replacement authenticators before the original authenticators expire.

There are different types of RSA SecurID authenticators including the following:

- RSA SecurID Card
- RSA SecurID Key Fob
- RSA SecurID PINPAD Card
- RSA SecurID Software Token

**NOTE - See the appropriate documentation included with your RSA Authentication Manager for specific instructions on using SecurID Authenticators.**

## User Passwords

A user password is a single password the user enters instead of a PIN and Tokencode. User passwords are less secure than other authenticator types, but they allow administrators to administer users with different security needs.

An administrator might assign user passwords to the following user groups:

- To employees who already work in a physically secure facility
- To new users to provide an initial level of protection while assigning more secure authenticators over a period of time
- If an employee loses or misplaces their hardware authenticator

In these cases, the administrator can manage all users from the same RSA Authentication Manager database. Because the user password is less secure than other authenticator types, RSA Security does not recommend user passwords as a long-term security solution.

## Two-Factor User Authentication

The authentication used by the RSA Authentication Manager is superior to traditional password authentication because it requires two factors instead of just one. To gain access to the protected system, users must enter a valid RSA SecurID Passcode made up of the following:

- Something you know: A secret, memorized personal identification number (PIN)
- Something you have: The current code generated by an authenticator assigned to the user

The first factor is something the user knows. The second factor is something unique and un-reproducible that the user possesses such a code from his or her RSA SecurID authenticator. Requiring both factors ensures exceptionally secure user authentication and access control.

The user password option offers only single-factor authentication and therefore is not as secure as the other RSA SecurID authenticators.

## RSA SecurID Tokencode Generation and Synchronization

RSA Authentication Manager software and RSA SecurID authenticators work together to authenticate user identity. The RSA Security patented time synchronization ensures that the pseudorandom code displayed by a user's authenticator is the same code the RSA Authentication Manager software has generated for that moment.

An RSA SecurID authenticator generates codes with a calculation that employs the following:

- The authenticator's unique identifier (also called a "seed"), which is stored in the authenticator itself and represented by the serial number on the back of the authenticator
- The current time according to the authenticator's internal clock, expressed in Coordinated Universal Time

### RSA Security's proprietary algorithm

The RSA Authentication Manager generates codes for an authenticator employing the following:

- The authenticator's unique identifier, which is stored in the authenticator's record in the RSA Authentication Manager database
- The time, which is calculated by adding the offset stored in the authenticator record to the current RSA Authentication Manager time, expressed in Coordinated Universal Time

The RSA Authentication Agent software uses a one-way hash to ensure that no one can see the PIN or Tokencode during transmission.

To determine whether or not an access attempt is valid, the RSA Authentication Manager compares the code it has generated with the code a user enters as this user's current RSA SecurID Tokencode. If the RSA SecurID Tokencode does not match or if the wrong PIN is entered, the user is denied access.

### Accountability and Security Auditing

Because user accountability is a critical part of system security, the RSA Authentication Manager creates an audit trail. This audit trail tracks all login requests and operations performed with the Database Administration application.

When the RSA Authentication Manager is properly implemented, the audit trail reliably identifies which user was responsible for each logged action. User information that is based on two-factor authentication provides stronger legal evidence of who performed the recorded activity than information based solely on password authentication.

Users identified by the RSA Authentication Manager audit trail cannot disown responsibility for security breaches perpetrated under their identities.

You can examine the audit trail in the following ways:

- Through Database Administration application reports.
- Through the Report Creation Utility.
- Through reports created with third-party software using the file generated by the automated log maintenance feature.
- By requesting that records be displayed on-screen as soon as they are created.

**Note:** Detailed information regarding RSA Authentication Manager reporting capabilities can be found in the administration documentation included in your RSA Authentication Manager Software.

## Protection from Intruders

If an unauthorized person tries to use a stolen PIN or RSA SecurID authenticator to break into your system, the RSA Authentication Manager “evasion-of-attack” features can detect the attempted intrusion and deny access.

Before learning more about these features, note the following points:

- Evasion-of-attack features do not replace the need to implement and use the product properly.
- Evasion-of-attack features can offer no protection against an intruder who has both a user's PIN and RSA SecurID token.

Therefore, it is essential that:

- all users protect the secrecy of their PINs and the physical security of their authenticators
- administrators respond immediately to disable compromised PINs and missing authenticators
- RSA Authentication Manager primary and replica servers be kept physically secure

## Evasion-of-Attack Features

Here are the two most important evasion-of-attack features:

### **A Stolen PIN with a Guessed RSA SecurID Tokencode Is Ineffective.**

If an unauthorized person with a stolen PIN eventually succeeds in guessing a valid RSA SecurID Tokencode, the person is not granted access because the system prompts for a second Tokencode after a series of failed login attempts. If the person does not enter the NEXT Tokencode generated by the authenticator, he or she is denied access.

### **Stolen RSA SecurID Authenticators are disabled automatically.**

If the RSA Authentication Manager detects repeated login attempts with valid Tokencodes but with PINs that are not valid for the authenticator, the system assumes that an unauthorized person has obtained the RSA SecurID authenticator and is using it with guessed PINs. The system disables that authenticator automatically and logs the event.

## Detecting a Replay Attack

The RSA Authentication Manager software warns you of any change in system time that may indicate a replay attack. In a replay attack, an intruder attempts to gain access with a captured Passcode by setting the server system clock back, then reusing the Passcode at the appropriate system time.

When the RSA Authentication Manager software detects that the server system clock has been set back, it puts the following warning message in the log database: \*\*\* System clock setback detected. This message can be viewed through Database Administration application Activity or Exception reports. This message is also added by default to the Event log and can be tracked and identified with a commercial network management tool. Because this message may indicate a serious security breach, RSA Security recommends that it not be removed.

# Getting Started

## Points of Integration

Interoperability or integration with the RSA SecurID authentication technology can be achieved in several ways:

**RSA Authentication Agent:** Partner product authenticates users with username and Passcode via RSA's native SecurID protocol and may include leveraging an existing RSA Authentication Agent for SecurID protection (i.e. – existing web or windows agent)

**RADIUS:** Partner product authenticates users with username and Passcode via standards-based RADIUS protocol.

**RSA Authentication Manager Administrative API:** Partner product provisions for and/or manages the Authentication Manager's user database via the RSA Authentication Manager.

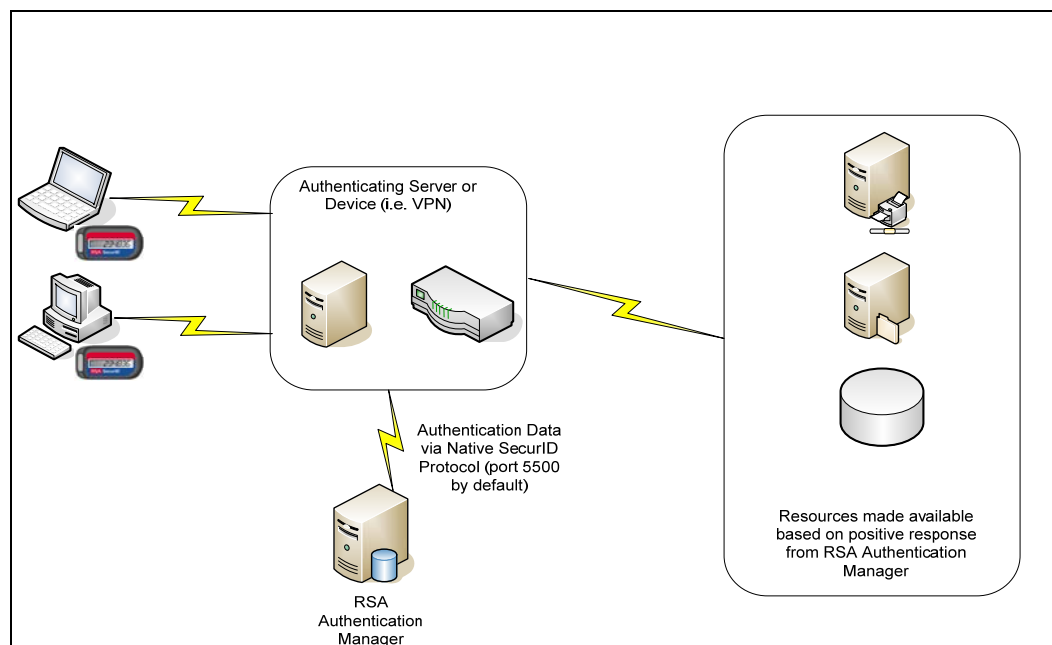
**RSA Software Token API:** Partner product leverages RSA Security's Software Token API to extract the Tokencode from the seed, either on the local pc or smartcard, and prompt the user for a PIN. This is often referred to as Software Token Automation.

## Planning the Partner Integration

Before you begin any development and/or testing, you should consult with your RSA Partner Engineering representative. This engineer will help you determine the correct approach, discuss alternatives and provide valuable insight to integration.

During the initial consultation, the assigned Partner Engineering representative will ask for a description of the existing solution and common deployment scenarios in which it is used.

## RSA Authentication Agent Overview



## Planning RSA SecurID Integration

The Authentication Agent code that implements RSA SecurID authentication is available through an application programming interface (API), or as a full set of source code (available in C). The developer has several options for implementing RSA SecurID authentication, depending on the scope and target of the work.

If the product or application that is being made RSA SecurID Ready operates in an environment for which RSA Security provides Authentication Agent code, the RSA Security-developed API libraries can be used. These libraries are available for many UNIX operating systems and Windows. See the Authentication Manager README file for specific operating systems and versions currently supported.

If the target of the work is an operating system for which RSA Security does not provide agent code, or for a proprietary environment such as a hardware or firmware-based product, Authentication Agent source code is available for porting. Note that source code is only supplied if required by the partner and if the partner has completed the appropriate agreements with RSA Security.

RSA Security's Partner Engineering Group is available for consultation and to assist developers in choosing the correct options and implementation for their specific product.

## Obtaining the Latest RSA SecurID Authentication Agent API's

Partners may download the latest API libraries and documentation from the following URL:  
<https://knowledge.rsasecurity.com>

## RSA Authentication Agent Overview

This section provides a brief explanation of the messages that the RSA Authentication Agent should display to users. To maintain consistency in all RSA Authentication Agent implementations, RSA Security strongly recommend using the text strings exactly as they are presented.

## RSA Authentication Agent Functionality and Messages

RSA Authentication Agents and RSA SecurID Ready applications are expected to support the following functions:

- Obtain user authentication information and relay it to the RSA Authentication Manager
- Relay the results of the RSA Authentication Manager's validation processing to the user
- Support New PIN operations
- Support NEXT Tokencode operation
- Support RSA Authentication Manager Replica Functionality
- Support Name Locking

The prompts and messages that the client displays to the user as part of the first four functions are described in the following text. Support of RSA Authentication Manager Replica functionality and name locking is transparent to users.

All functionality must be implemented. An inability to support any of these functions should be brought to the attention of RSA Security promptly. Failure to support all of these features may result in the lack of consistency across the customers' protected environment as well as loss of compatibility with RSA Security's own RSA Authentication Agents and decreased ease of use for users.

## The “Enter Passcode”: Prompt

The RSA SecurID authentication challenge is Enter Passcode. Users must respond by entering their RSA SecurID Passcode, comprised of their secret PIN (Personal Identification Number) and the Tokencode currently generated by the user’s RSA SecurID authenticator. Users with RSA SecurID PINPAD Cards enter the PIN into the PINPAD card itself; the result that is displayed on the user’s authenticator is the complete Passcode that must be entered as displayed. The Enter Passcode prompt is normally displayed in lieu of an “enter password” prompt.

## RSA Authentication Manager Replica(s)

The optional RSA Authentication Manager replica(s) run on separate machines and can be used to distribute authentication. The RSA Authentication Manager replica(s) is in regular communication with the RSA Authentication Manager via a dedicated TCP/IP socket. The API provides transparent failover to any available primary/replica. The RSA Authentication Manager replica does not have administrative capabilities. All RSA Authentication Manager Administration must be done on the primary server.

## Passcode Accepted

This message is displayed on the user’s screen after the user has entered a valid Passcode. The user has successfully authenticated and now has access to the RSA Authentication Manager-protected resource.

## Access denied

This message is issued by the RSA Authentication Agent to indicate a failed authentication request (for example, an invalid Passcode or the user is not activated on the client). The individual is denied access to the RSA SecurID-protected system. After an Access denied message is displayed, the user may be prompted again with Enter Passcode.

This message may be displayed for any of the following reasons:

- The RSA SecurID Card or RSA SecurID Key Fob user entered a valid PIN followed by an invalid Tokencode. The code could have been invalid because it was used previously, because it was mistyped, or because an unauthorized user guessed it that did not have the token.
- When using an RSA SecurID PINPAD Card, the user entered an invalid Passcode. The code could have been invalid because it was used previously, because it was mistyped, or because an unauthorized user guessed it that did not have the token.
- The RSA SecurID Card or RSA SecurID Key Fob user entered an invalid PIN followed by a valid Tokencode. The PIN could have been invalid because it was mistyped, guessed, or the authenticator was in New PIN mode and its previous PIN had been cleared.
- When using an RSA SecurID PINPAD Card, the user entered an invalid PIN into the card, and therefore, an invalid Passcode was generated.
- The user’s RSA SecurID authenticator is disabled. Tokens can be disabled either automatically to evade a system attack or by an administrator.
- A person attempting to gain unauthorized access is guessing Passcodes.
- The user is not activated on the client directly or via group membership.
- The client was not found in the RSA Authentication Manager database.
- Mismatch of node secret or encryption type.
- Authenticator has expired or the user’s temporary access period has expired.

See the appropriate Authentication Manager Administration documentation for more information on the different types of user authentication failures.

**Important: When developing the partner solution, do not communicate the reason for the denial to the end user. The Partner Solution should only provide the user with the message “Access Denied” if the authentication has failed.**

Do not disclose any information regarding the type of security system and do not identify which aspect of the login attempt failed. Display only Access denied. Providing more than minimal information to potential hackers can facilitate future attacks on the protected resources.

More detailed information about the unsuccessful login attempt is available to the RSA Authentication Manager administrator in the RSA Authentication Manager audit trail.

## New PIN Mode

When an RSA SecurID authenticator is first assigned to a user, a PIN is not yet associated with it. If the “PIN-Less” token functionality is not utilized on the Authentication Manager, the RSA SecurID authenticator cannot be used for authenticating until its assigned user performs the New PIN operation. Alternatively, an authenticator can be put into New PIN mode by an administrator at any time.

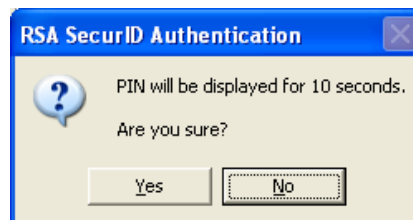
When an RSA SecurID authenticator is in New PIN mode, it cannot be used for authenticating until its assigned user performs the New PIN operation unless “PIN-Less” token functionality is utilized on the Authentication Manager. During the New PIN operation, the RSA Authentication Manager will either assign a PIN to the user or allow the user to specify a PIN that he or she will use. Which options are displayed to the user who initiates a New PIN operation depends on how the RSA Authentication Manager is configured. For information on setting this parameter, see the appropriate RSA Authentication Manager Administration documentation.

**Note:** If New PIN procedure is canceled, ACCESS DENIED should be displayed.

## System Generated PIN

New PIN Operation: User receives a PIN

Prompt when user’s RSA SecurID authenticator is in New PIN mode and the user initiates a new PIN session.



## User-Selectable PIN

New PIN Operation: User may either specify or receive a PIN

Prompt when the user's RSA SecurID authenticator is in New PIN mode and the user has initiated a session. This prompt indicates that the RSA Authentication Manager will generate a new PIN for the authenticator or allow the user to create his or her own PIN.



The screenshot shows a 'New PIN' dialog box with the RSA SecurID logo and the text 'A new PIN is required!'. There are two radio buttons: 'RSA ACE/Server will generate PIN' (which is selected) and 'I will create PIN'. Below the second option is a text input field for the PIN and a 'Confirm' button. The dialog has 'OK' and 'Cancel' buttons at the bottom.



The screenshot shows the same 'New PIN' dialog box, but the 'I will create PIN' radio button is now selected. The text input field for the PIN and the 'Confirm' button are visible below it. The 'OK' and 'Cancel' buttons remain at the bottom.

## User-Defined PIN

New PIN Operation: User specifies a PIN

Prompt when the user's RSA SecurID authenticator is in New PIN mode and the user has initiated a session. This prompt indicates that the user will specify a new PIN for the authenticator.



This screenshot is identical to the one in the 'User-Selectable PIN' section, showing the 'New PIN' dialog box with the 'I will create PIN' option selected. It includes the RSA SecurID logo, the message 'A new PIN is required!', the selected radio button, the PIN input field, the 'Confirm' button, and the 'OK' and 'Cancel' buttons at the bottom.

**Note:** The PIN length and PIN type are components configured within the RSA Authentication Manager.

## PIN Rejected

The PIN Rejected message appears when the user has selected an unacceptable PIN. The PIN specified by the user must conform to the system PIN specifications for length and allowable characters (alpha-numeric combinations). PIN numbers must be composed of digits (0-9) or letters (A-Z). RSA SecurID PINPAD Cards require that the PIN be composed exclusively of digits (0-9), the PIN cannot begin with zero, and the PIN length cannot exceed the length of the Tokencode.

**Note:** The Agent should be designed to validate that the new PIN entered by the user conforms to the required PIN characteristics set by the RSA Authentication Manager administrator.

## Re-authentication

After the PIN has been set, it is necessary to re-authenticate the user. They should be instructed to wait for the Tokencode to change and to authenticate with the Passcode.

**Note:** When designing the agent, do not call AceClose after setting the PIN. This will ensure you re-authenticate to the Authentication Manager that holds the new PIN information.

## Next Tokencode Mode

When an authenticator is in Next Tokencode mode and it is used in a login attempt, the user is required to input a second successive Tokencode from the RSA SecurID token. The RSA Authentication Manager puts an authenticator into Next Tokencode mode if the authenticator has drifted out of synchronization with the server system's clock or if it seems that the token's PIN has been compromised and an unauthorized user is attempting to guess a valid Tokencode. Requiring two consecutive Tokencodes ensures that the user actually has the RSA SecurID authenticator in his or her possession.

RSA Authentication Agents must be able to perform the Next Tokencode dialog so that the RSA Authentication Manager can properly identify users whose RSA SecurID authenticator have drifted out of sync with the RSA Authentication Manager and so that the RSA Authentication Manager can take evasive action when a compromised PIN appears.


## SecurID Integration for RADIUS Server Products

Unlike the integration between RSA Authentication Manager and RADIUS clients, integration with RSA SecurID involves embedding RSA's native authentication API into the partner's RADIUS server. The RADIUS server now becomes an agent to the RSA Authentication Manager. In this scenario, authentication requests are made to the RADIUS server via the standards-based RADIUS protocol. The RADIUS server then provides the translation to RSA's Authentication Manager via RSA's native protocol. As such, the RADIUS server must correctly implement the RADIUS challenge-response protocol with the RADIUS clients to facilitate the additional prompting necessary for users to enter New Pin and Next Tokencode modes.

## SecurID with Web Based Applications

The Web access authentication (formerly known as "WebID") feature set of the RSA Authentication Agent software uses RSA SecurID authentication to protect selected virtual servers, directories, and files on specified web servers on selected platforms (See product documentation on [www.rsasecurity.com](http://www.rsasecurity.com) for details on supported platforms and servers).

When you enable Web access authentication to protect Web server resources, users who attempt to access the protected resources are prompted for their RSA SecurID Passcode. Users



who enter a valid Passcode are allowed access to the protected pages. Users unable to provide a valid RSA SecurID Passcode are denied access.

Only RSA SecurID users who are registered in the RSA Authentication Manager databases can access Web resources that are protected by RSA SecurID. As a result, you can use your Web site as both a public resource available to all users and a highly secure area for posting confidential information to trusted users.

Two ISAPI filters determine which users get access to a protected resource. One filter screens URL's to determine if the resource is protected. The second checks the cookie to see if the user has authenticated and also checks the state of the session to determine if re-authentication is required.

## **How to incorporate Web Access Authentication**

Partners can provide RSA SecurID authentication to protect web resources in one of two ways:

- by either utilizing the RSA Authentication Agent for Web Servers
- by building the RSA Authentication Agent API's into an application

## **Leveraging the RSA Authentication Agent**

The RSA Authentication Agent can be leveraged to determine the web resources that are protected as well as determine if a user has previously been authenticated.

## **Building in the API's**

Fully integrating the API's into a Partners application can determine which web resources are protected. The configuration must be done from the administration page of the application's policy manager.

## **Distributing Web Access Authentication Cookies**

When you enable the Web access authentication feature set on a virtual Web server, you automatically enable Web access authentication cookies for that server. Each time that an RSA SecurID user enters a valid Passcode at the Web access authentication prompt, the web server gives the user a Web access authentication cookie. The cookie, which is stored in the user's Web browser, passes the user's authentication information to the server when the user browses to a protected file or directory on that server. As long as the cookie's validity has not expired, the user is prompted only once for a Passcode in the current session.

For the Web Access Authentication Cookies to work, RSA SecurID users must enable the cookie acceptance feature in their browsers. They must also use Web browsers that support Forms and Persistent Client State HTTP Cookies. See the appropriate RSA Authentication Agent documentation for information about supported browsers.

If you want the virtual server to issue cookies that are valid on multiple servers in the same domain or across multiple Web domains, you can enable domain cookies. See "Enabling Domain Cookies" for information about enabling this feature.

## Setting Cookie Expiration Times

A Web access authentication cookie is valid only during the browsing session for which it was created. If the user exits the Web browser, the cookie expires, and the user must get a new cookie during the next authentication session.

Before enabling Web access authentication cookies, decide what expiration constraints, if any, to place on the cookies you distribute to RSA SecurID users. For example, you can configure cookies to always expire during a browsing session. Setting an expiration time can help reduce the likelihood that an unauthorized user will gain access to protected pages through a Web browser left unattended by a trusted user.

**Note:** To increase the effectiveness of idle cookies, instruct your RSA SecurID users to enable the option in their browsers that always forces the updating of pages. Doing so ensures that the cookies are refreshed, too. If the cookies are not periodically refreshed, the RSA Authentication Manager will be unable to update the cookies, so RSA SecurID users will be prompted to re-authenticate when the idle cookies' timeout period expires.

## Enabling Domain Cookies

By default, Web access authentication distributes cookies that are valid only on the Web server from which they are issued. To issue cookies that are valid on multiple servers in the same Web domain or across multiple domains, you can enable the Domain Cookie feature.

**Note:** Domain cookies bypass a workstation's client activations in the RSA Authentication Manager database. RSA SecurID users whose browsers use a domain cookie from one server might gain access to information on other servers that they are usually not allowed to view.

Use Windows file permissions in conjunction with RSA Authentication Manager grouping and client activation capabilities to configure your server or domain to be highly selective about which protected files it allows the RSA SecurID users to view.

For example, if you post your organization's quarterly sales performance figures in confidential URLs on a Web server named Sales, and the Sales server has domain cookies enabled, an RSA SecurID user with a domain cookie issued by the Engineering server can gain access to the Sales server and view the confidential URLs. To prevent this from happening, restrict access to the confidential directories by assigning Read permission only to the appropriate RSA SecurID users.

## Using the Web Access Authentication HTML Forms


During installation, the Agent setup program copies HTML forms, or templates, into the \aceclnt directory on the Agent host. The Agent uses these forms to display the Web access authentication prompt and its companion pages. The HTML forms that it installs are customizable with a basic HTML editor and can be configured to provide the appropriate look and feel to the authentication screen.

## Customizing Message Strings

You can customize certain messages that display while users interact with the Web access authentication prompt pages that are produced from the HTML forms. The message strings are contained in a file named strings.txt located in the \aceclnt directory.

Developing an application to take advantage of Active Directory domain password feature of the RSA Authentication Manager

With the introduction of the RSA Authentication Manager 6.0, a powerful tool is now available to applications that wish to leverage Microsoft Active Directory domain credentials within an RSA SecurID for Microsoft Windows deployment.



This new functionality securely stores Active Directory Domain credentials inside the RSA Authentication Manager database which can be retrieved via the RSA SecurID Authentication API. The new API functionality will allow the partner product to replay the stored credentials to resources within an RSA SecurID for Microsoft Windows environment.

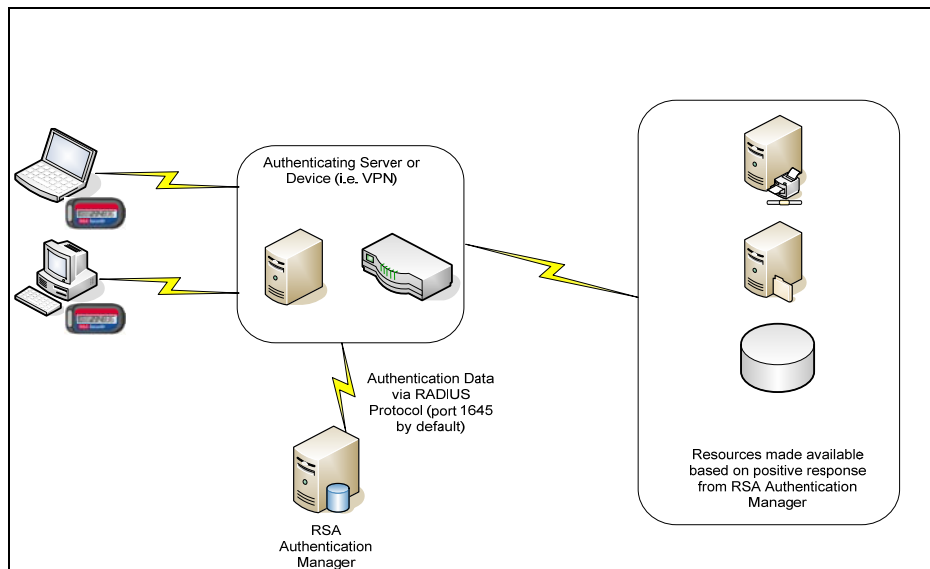
Functions available for this type of integration include the following:

- SD\_InitEx
- AceGetLoginPW
- AceSetLoginPW
- AceGetDAAuthenticationStatus
- AceGetDAAuthData

**NOTE:** These features should be implemented as an optional and configurable component as RSA SecurID for Windows is an essential component for these deployments.

Requirements for this feature must include an Active Directory environment that is protected by RSA SecurID for Microsoft Windows. By including this additional functionality, the partner product must also demonstrate that the cached domain credentials can be passed to a protected resource for authentication.

# RSA SecurID Authentication via RADIUS Protocol



Traditionally, in order to add RSA SecurID authentication into a hardware or software product, a vendor would utilize the RSA Authentication Agent code for integration into their product. By implementing RADIUS client code into their product, a company can be compatible with a variety of authentication schemes, including RSA SecurID authenticators, without having to add specific code to support them. This is done through the support of RSA SecurID authentication via the RADIUS protocol originating at an authenticating device or application.

The RSA Authentication Manager includes a RADIUS server that listens on port 1645 and 1812 by default. This is configurable through the Authentication Manager Configuration utility. The built in RADIUS server expects authenticating clients to support the challenge/response mechanisms built in to the RADIUS protocol in order to support New PIN and Next Tokencode modes.

Information about this protocol can be found from public working groups such as <http://ietf.org>. It is up to the partner to ensure that the RADIUS protocol has been correctly implemented.

In some cases, the RADIUS client requires a configurable timeout value. This often needs to be increased from the RADIUS default of 3 seconds due to the slight time required to perform an RSA SecurID authentication.

It is essential that the RADIUS client pass the Access-Challenge text directly to the user without checking. This prevents problems caused by localization of the prompts; if the RADIUS client is looking for specific prompts and these prompts have changed, New PIN and NEXT Tokencode may not work.

The following table maps the RSA native protocol to the appropriate RADIUS response.

RSA Authentication Manager Response	RADIUS Response
Passcode Accepted	Access-Accept
Access Denied	Access-Reject
New PIN Required	Access-Challenge
NEXT Tokencode Requested	Access-Challenge

# The RSA Authentication Manager Admin API

---

For those partners and customers that would like to perform administrative or reporting functions with the RSA Authentication Manager, the partner product may utilize an Administrative API which is located in the \ace\utils directory of the RSA Authentication Manager Server installation.

For additional information on functionality, please refer to the RSA Authentication Manager Administrative API documentation which can be found in the in the \ace\doc directory.

There are no minimum requirements for implementing administrative functionality, other than not causing ill effects to the RSA Authentication Manager database. Once the integration has been completed, the partner will be required to complete an implementation guide to describe the level of functionality and capabilities of the integration.

Please contact your RSA Partner Engineering representative to obtain the latest guide template.

Certification testing will be customized to focus on the specific functions that the partner product intends to provide. Please work with your assigned RSA Partner Engineering representative to coordinate testing for the partner product.

## Remote Management Capabilities

Partners frequently inquire about the remote capabilities of RSA Authentication Manager's administrative API. There are no remote capabilities to this interface and as such, it is entirely up to the partner to provide such capability if there is a need. In these cases, a module that utilizes the native communications protocol of the partner product (usually referred to as an administrative agent) is created to facilitate this functionality.

**Note:** There is no direct access to the underlying Progress Database within the Authentication Manager. All administrative functions must be through the exposed administrative application program interface.

Documentation is required in the implementation guide to demonstrate how the administrative agent is installed and/or distributed to the Authentication Managers. In a Windows environment, install packages and scripts should provide all registry edits as manual registry edits are not supported.

## The RSA SecurID Software Token API

---

The RSA SecurID Software Token is a software version of the RSA SecurID authenticator and has an application program interface to allow for the extraction of the Tokencode which can be used to authenticate via a client application on the users desktop.

In the case of a VPN client that utilizes this API, a user would only be prompted to enter his/her pin in order to authenticate to the device. RSA SecurID Software Token API integrations should never cache the user's PIN.

The RSA SecurID Software Token API and associated documentation can be found available on the RSA SecurID Software Token distribution media included in your partner shipment. This media will also contain the Software Token seed records, as well as the RSA SecurID Software Token application to be used with the RSA Authentication Manager during certification testing.

## Documenting the Solution

---

The RSA Secured Partner program's main goal is to provide our mutual customers with configuration support as well as a comfort level that the products will work together today as well as in the future. To provide this comfort level, RSA Security provides templates to be used in the creation of implementation guides that will be posted on [www.rsasecured.com](http://www.rsasecured.com). Please contact your Partner Engineering representative to get the latest template.

The template for RSA Secured solutions contains product and interoperability information that help our mutual customers understand and deploy the joint solution.

When documenting the solution, we will ask that you include screenshots to help visualize the integration steps. While images can be very helpful in a technical document, try to keep your screenshots cropped to a size where they are both readable and functional.

Keep in mind that screenshots should only be used to help describe the UI of an application. Screenshots are not intended to replace good documentation or multiple configuration steps.

Also, when possible try to use GIF or PNG images as the file compression and quality will make the overall document more manageable for end users.

## How do I fill out the RSA SecurID Implementation Guide?

### Partner Information

For this section, include all relevant information regarding the Partner organization including the Company Name, Corporate Web Site Address, Product Name and Version and Platform. For the Product Description heading, include a brief paragraph that will serve as an overview of the Partner Product itself.

The Product Category heading is used to list the Partner Product on the RSA website. The categories can be found at: <http://www.rsasecured.com>. If you are unsure of which category the product belongs in, please ask your Partner Engineering representative.

### Solution Summary

For this section, include a short paragraph summarizing the integration between the partner product and RSA Authentication Manager.

It is helpful to explain an overview of how the integration works. This can be done by describing the typical deployment, a business case that this integration solves, or by including a list of benefits from a joint solution.

A diagram of a typical deployment scenario should also be included in this section.

## Partner Integration Overview

In this section, a table is provided to help outline specific details of the integration with the partner product. These items are broken down as follows:

**Authentication Methods Supported:** This lists the protocol(s) by which a partner product communicates with the RSA Authentication Manager.

**Library Version Used:** This only applies to partner products that communicate using the Native RSA SecurID Protocol. The partner should list the version of the .dll or .so library used. If source is used, contact your Partner Engineering representative to provide this information.

**RSA Authentication Manager Name Locking:** This is functionality included in the native RSA SecurID protocol. When using Native RSA SecurID Protocols, this functionality is mandatory for certification.

**RSA Authentication Manager Replica Support:** Ability to support up to 10 RSA Authentication Manager Replicas via the Native RSA SecurID protocol. When using Native RSA SecurID Protocols, this functionality is mandatory for certification.

**Secondary RADIUS Server Support:** When using the RADIUS protocol, this defines the ability of the partner product to make requests to multiple RSA Authentication Manager Servers. Please provide the number of servers supported.

**Location of Node Secret on Agent:** When using the Native RSA SecurID protocol, a file called the Node Secret is used to encrypt communication. Enter the location where the Node Secret is stored. (e.g. C:\WINDOWS\System32)

If the product uses RADIUS or does not store the Node Secret, then indicate None Stored.

**RSA Authentication Agent Host Type:** The partner product must be defined as an Agent Host in the RSA Authentication Manager Database. Define the Agent Host type that is used to configure communication with the partner product. This can be only one of the following: Communication Server, Net OS Agent, Net SP Agent, or UNIX Agent.

**RSA SecurID User Specification:** This item outlines how RSA SecurID authentication is used to protect the users of a partner product.

- If RSA SecurID authentication is enabled for individual users or groups, choose Designated Users.
- If RSA SecurID authentication is configured to be on or off for all users, choose All Users.
- If RSA SecurID authentication is the default method, then choose Default Method.

**RSA SecurID Protection Administrative Users:** If the product can be utilized to authenticate administrators of the partner product using an RSA SecurID authenticator, indicate Yes. Otherwise, select No.

**RSA Software Token API Integration:** If the partner product can be configured to integrate with RSA Software Token automation then indicate Yes. If the partner product does not use these API functions select No.

If the partner product uses the RSA Software Token API as an integration point, there are additional items in the checklist that will be mandatory for certification.

**Use of Cached Domain Credentials:** If the partner product can be configured to utilize the RSA SecurID API Cached Domain Credential functionality then indicate Yes. If the partner product does not use these API functions, or is using RADIUS, select No.

If the partner product is designed to use the Cached Domain Credential functionality, there are additional items in the checklist that will be mandatory for certification

## Product Requirements

The tables provided in this section may be replaced with a pointer or reference to partner product documentation or read-me files if necessary. In the case that this pointer is added in place of system requirements, the provided tables should be deleted from the finished template.

### Partner Product Requirements:

In this section, please list the minimum requirements for all partner product components. You may need to document multiple components in the case where a server and client application are used together in the integration.

If a particular item or section does not apply to this particular product, please delete the corresponding section. Please refer to the table below for an example of how this information should be entered.

Partner Product Requirements: <Partner Product (Component)>	
CPU	
Memory	
Storage	
Firmware Version	

### Operating System Support:

Please list all applicable operating systems that are supported by this Partner product. Only list product versions that are either required or provide support for the integration.

Also, be sure to include required patch or service pack information as required. If no specific service pack is required, please enter "All Patch Levels Supported" in the given field.

Operating System	
Platform	Required Patches

### Additional Software Requirements:

Please list all additional software requirements in this section. This includes but is not limited to Third Party LDAP Servers, Specific Web Browsers and Java Runtime Environments. Also list required patches or service packs as required in the appropriate fields.

Additional Software Requirements	
Application	Additional Patches

Note: If hot-fixes or patches are required to configure the partner product for interoperability, you must also document this information in the section labeled "Known Issues". The information on how the patch is identified and obtained must also be included in the "Known Issues" section as well.

## RSA Authentication Manager Configuration

List the basic steps required to configure the partner product as an Agent Host in the RSA Authentication Manager's Database Administration.

- Replace the term <PARTNER PRODUCT> with the actual name of the product.
- For Agent Type, define the setting for the Agent Host used in your integration testing.
- If the partner product is a RADIUS client, explain that extra steps are necessary to add an encryption key for the Agent Host record.

## Partner Authentication Agent Configuration

When documenting the solution, the partner should assume that the RSA Authentication Manager environment is already up and running, and also that the audience has some basic familiarity with all of the products involved. The Implementation Guide is intended to fill in the gaps between the Partner Documentation and RSA Authentication Manager Documentation.

If appropriate, you may also wish to include a note explaining that these procedures should not be attempted until certain prerequisites have been met. For example:

“Before attempting to enable RSA SecurID authentication on the partner product, ensure that you have properly installed the RSA Authentication Agent for Windows and that you can successfully perform a test authentication to the RSA Authentication Manager.”

“Please be sure that your RSA Authentication Manager is both configured for and able to authenticate via RADIUS before beginning the integration.”

## Integration Overview

It may be helpful to outline all of the steps involved in the integration before you explain each one. This overview should provide basic task information regarding the steps you will complete, and each task set should provide headings and footnotes where applicable.

If there are any “add-on” modules or administrative software to be installed before RSA SecurID authentication can be enabled, then it should also be documented here.

If an administrator would need any special rights to install or configure this software then be sure to document this. Be sure to include screenshots and directions on how to install and configure the software if it is not mentioned in the partner product documentation.

## Documenting the Integration

List detailed steps required to configure the product for RSA Authentication Agent operation. If the product supports multiple methods of authentication (Native RSA SecurID Authentication Protocol, RADIUS, etc.) describe the steps required to enable each type of authentication in separate sections.

In covering the procedure, you should address such questions as:

- How do you turn on RSA SecurID authentication in the Client?
- What steps are required to configure the Client to recognize the IP address and UDP port of the RSA Authentication Manager and Replicas?
- What configuration options are enabled on remote users' software (if applicable)?
- If the product maintains a user database separate from the RSA Authentication Manager database, how do you add users and enable them for RSA SecurID authentication?
- How do you clear the Node Secret on the product?
- Provide examples of RSA Authentication Agent logon screens.
- How does the partner product utilize the Domain Password retrieved from the RSA Authentication Manager?

Also be sure to include screenshots from the end-user experience that show a standard authentication, a new pin dialog, and a Next Tokencode screen. Including a description of what

the user should experience after a successful authentication, as well as when access has been denied is also recommended.

## Certification Environment

List all components used during the testing for certification. All major components in the certification environment should include the product version information as well as the operating system they were installed on.

Certification Environment		
Product Name	Version Information	Operating System
RSA Authentication Manager		
RSA Authentication Agent		
RSA Software Token		
<Partner Product>		

## Certification Checklist

This checklist must be completed during certification testing. Do not remove any items from this checklist. If the selection does not apply, please list N/A in the appropriate field. Use images listed in the key on the bottom of the form to reflect Pass / Fail results.

Partners should test functionality based on the protocols that are supported by the partner product. In the case that both Native RSA SecurID Authentication and RADIUS protocols will be certified, all tests must be completed using both protocols.

## Mandatory Functionality

This portion of the checklist outlines the functions that are mandatory for a partner product to support in order to pass certification.

If a particular item in this checklist is not supported by the partner product by design, an exemption must be requested through your assigned Partner Engineering representative.

Mandatory Functionality			
RSA Native Protocol		RADIUS Protocol	
<b>New PIN Mode</b>			
Force Authentication After New PIN	<input type="text"/>	Force Authentication After New PIN	<input type="text"/>
System Generated PIN	<input type="text"/>	System Generated PIN	<input type="text"/>
User Defined (4-8 Alphanumeric)	<input type="text"/>	User Defined (4-8 Alphanumeric)	<input type="text"/>
User Defined (5-7 Numeric)	<input type="text"/>	User Defined (5-7 Numeric)	<input type="text"/>
User Selectable	<input type="text"/>	User Selectable	<input type="text"/>
Deny 4 and 8 Digit PIN	<input type="text"/>	Deny 4 and 8 Digit PIN	<input type="text"/>
Deny Alphanumeric PIN	<input type="text"/>	Deny Alphanumeric PIN	<input type="text"/>
<b>Passcode</b>			
16 Digit Passcode	<input type="text"/>	16 Digit Passcode	<input type="text"/>
4 Digit Password	<input type="text"/>	4 Digit Password	<input type="text"/>
<b>Next Tokencode Mode</b>			
Next Tokencode Mode	<input type="text"/>	Next Tokencode Mode	<input type="text"/>
<b>Load Balancing / Reliability Testing</b>			
Failover (3-10 Replicas)	<input type="text"/>	Failover	<input type="text"/>
Name Locking Enabled	<input type="text"/>	Name Locking Enabled	<input type="text"/>
No RSA Authentication Manager	<input type="text"/>	No RSA Authentication Manager	<input type="text"/>

## Additional Functionality

This section is only required only when specific RSA API's have been included in the partner integration.

Additional Functionality			
<b>RSA Software Token API Functionality</b>			
System Generated PIN	<input type="text"/>	System Generated PIN	<input type="text"/>
User Defined (8 Digit Numeric)	<input type="text"/>	User Defined (8 Digit Numeric)	<input type="text"/>
User Selectable	<input type="text"/>	User Selectable	<input type="text"/>
Next Tokencode Mode	<input type="text"/>	Next Tokencode Mode	<input type="text"/>
<b>Domain Credential Functionality</b>			
Determine Cached Credential State	<input type="text"/>	Determine Cached Credential State	<input type="text"/>
Set Domain Credential	<input type="text"/>	Set Domain Credential	<input type="text"/>
Retrieve Domain Credential	<input type="text"/>	Retrieve Domain Credential	<input type="text"/>

INIT / PAR

✓ = Pass ✗ = Fail N/A = Non-Available Function

When either the RSA Software Token API Functionality or Domain Credential Functionality has been included the associated checklist items become mandatory. If these additional features are not included in the partner product, please list N/A to signify non available functionality.

## Known Issues

List any known issues or problems that have been discovered during development or certification testing. This section should reference required hot-fixes or patches, known configuration issues and viable workarounds to common issues.

If the hot-fix has been issued by the 3rd party partner, this information should also be included in the Partner Product Requirements section with a pointer to the Known Issues section. Please include the information on how the hot-fix or patch is obtained, as well as all relevant information to identify if the patch has already been installed.

Some examples of information contained in this section include the following:

### **Known permissions issues with Registry Keys and Values:**

In order to assure that the partner application has full access to read or write node secret data. The administrator must modify permissions on the Windows Registry Keys to allow the <Partner Service> access to this information.

### **Known issue with interoperability**

In order to correctly interoperate with the <product>, a hot-fix has been released to correct known issues with this integration. (Reference hot-fix or patch number and the method in which a customer is to obtain this patch.)

This section is included as an additional resource to customers and should contain information that is helpful to this particular integration only. While references to provided material or web articles are acceptable, do not include information that has already been provided in the associated partner or RSA product documentation.

## Appendix

This section is provided for documentation of issues or topics that are outside the scope of the included sections. This section is optional and should be deleted if it is unused.

- Some examples of information contained in this section include the following:
- Define any technologies or terms specific to the Partner product.
- References to technical documentation (e.g. RFC, Solution Overviews or White Papers, etc...)
- Additional information regarding special configuration of required components.

# Completing RSA Secured Certification Testing

---

This portion of the document describes how the partner should prepare for and perform the tests necessary to complete the RSA SecurID Ready checklist. These steps are listed in order and should guide you through the testing procedures in an efficient manner.

To document your results, you should print out the RSA SecurID Ready Checklist which is located at the end of the RSA SecurID Ready Implementation Guide template.

During the completion of the testing, you will be asked to perform various tasks related to the RSA Authentication Manager configuration; in addition you will also need to create user and Agent Host records within the RSA Authentication Manager database. While it is the intention of this document to outline steps to help simplify the testing procedures, this document will not provide detailed information on many of these tasks.

For additional information on any administrative functions of the RSA Authentication Manager, please refer to RSA Administration guides included within the installation directory of the product.

## The Environment

The tasks to complete certification testing require that you have installed at least three RSA Authentication Manager Servers. You should have one Primary Server and a minimum of two Replicas configured before you begin the certification testing.

The Partner Product should be installed and communicating on the same network segment as the RSA Authentication Manager Servers. The reason for this requirement is to alleviate unnecessary variables in the event of authentication failure. This network model will also simplify any troubleshooting that may be required as testing goes forward.

While customer environments will undoubtedly be far more complex than a single network segment, the intention of the certification testing is to prove interoperability in its most basic form and to allow the configuration steps to be documented for future installations.

## Problems, Questions and Failures

Any issues encountered during certification testing that prove difficult to troubleshoot should be reported to your designated Partner Engineering representative.

In the event that you need assistance with the certification testing, your Partner Engineering representative may be able to provide access to a group of RSA Authentication Manager Servers which can be used over the WAN for remote testing.

## Using the RSA Authentication Log Monitor

During the certification you will want to view the results of the authentication requests. The Log Monitor is provided as a reporting tool for all authentication requests made to the RSA Authentication Manager.

1. To start the Log Monitor on the Primary Authentication Manager or on a Replica, navigate through the Start Menu to **Programs | RSA ACE/Server | Log Monitor**.
2. Select the default filter settings and click OK.

## Initial Configuration

On the Primary RSA Authentication Manager, open the Local Database Administration tool by navigating through the Start Menu to Programs | RSA ACE/Server | Database Administration – Host Mode.

### Create Test User Entries

3. Create three user records with names that will be distinguishable for testing purposes.
4. Assign each user a token and a user Password as listed in the table below.

User 1 : Set for system-generated PIN.  
"Allowed to create a PIN" and "Required to create a PIN" unchecked.

User 2 : Set for user-selectable PIN.  
"Allowed to create a PIN" checked and "Required to create a PIN" unchecked.

User 3 : Set for user-defined PIN.  
"Allowed to create a PIN" and "Required to create a PIN" checked.

Serial Number	Token Type/Auth With	Status
000026354848	Standard/Passcode	Enabled;New PIN Mode
000025530619	PINPAD/Passcode	Enabled;New PIN Mode
-----	Password	Enabled;Change Required

O: Original token R: Replacement for previous token

**Figure 1 – Edit User Screen**

5. Change the System Parameters to provide constraints around PIN length and format. To change these settings, select System | System Configuration | Edit System Parameters from the RSA Authentication Manager Database Administration menu bar.

Verify that the following items are set as follows.

- Min PIN to 4.
- Max PIN to 8.
- Check the box for Alphanumeric PIN Allowed.

## Create Agent Host Entries

1. To create an Agent Host record for the partner product, select Agent Host | Add Agent Host.
2. For a new Agent Host, the Node Secret Created box will be disabled by default.
3. Ensure that Open to All Locally Known Users is checked.
4. Ensure that the Requires Name Lock is checked.
5. Ensure that Enable Offline Authentication is not checked.
6. Ensure that Enable Windows Password Integration is not checked.
7. Leave all other configuration settings as default.
8. On the partner product, ensure that there is not an existing Node Secrets established or stored. If one exists, make sure to clear the Node Secret before you begin the testing.

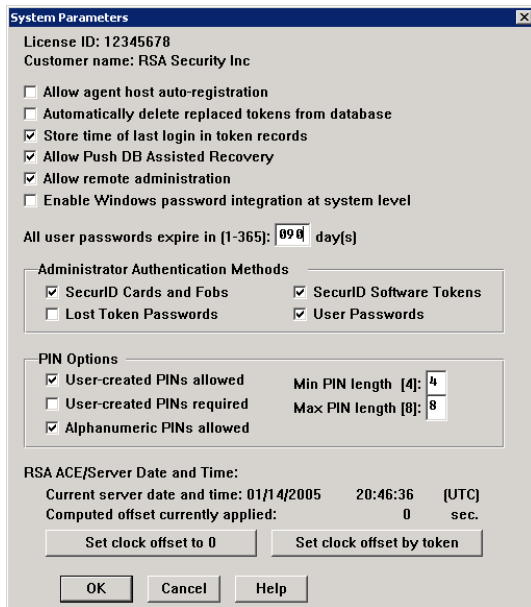
The screenshot shows the 'Add Agent Host' dialog box. The 'Name' field is filled with 'partnerProduct.yourDomain.com'. The 'Network address' field is filled with '10.100.50.42'. The 'Site' field is empty, with a 'Select' button next to it. The 'Agent type' dropdown is set to 'UNIX Agent'. The 'Encryption Type' has radio buttons for 'SDI' and 'DES', with 'DES' selected. Below these are several checkboxes: 'Node Secret Created' (disabled), 'Open to All Locally Known Users' (checked), 'Search Other Realms for Unknown Users' (unchecked), 'Requires Name Lock' (checked), 'Enable Offline Authentication' (unchecked), 'Enable Windows Password Integration' (unchecked), and 'Create Verifiable Authentications' (unchecked). At the bottom, there are three main buttons: 'OK', 'Cancel', and 'Help'. Above these are two columns of buttons: the left column includes 'Group Activations...', 'Secondary Nodes...', 'Edit Agent Host Extension Data...', and 'Assign Acting Servers...'; the right column includes 'User Activations...', 'Delete Agent Host', 'Assign/Change Encryption Key...', and 'Create Node Secret File...'.

**Figure 3 – Editing an Agent Host**

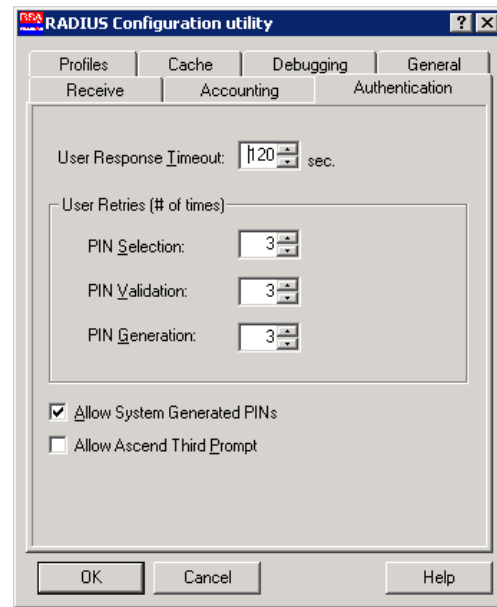
**Note:** In the event that the partner product will utilize the Cached Domain Credential function of the RSA Native Authentication API, the item “Enable Windows password integration” must be checked.

## System Generated PIN Configuration (RADIUS ONLY)

In the case where the partner product will be authenticating using the RSA Authentication Manager RADIUS listener, you must also enable the transmission of System Generated PIN's using the RADIUS Configuration Utility.



**Figure 2-A – Edit System Parameters**



**Figure 2-B – RADIUS Configuration**

1. From the Start Menu, select Programs → RSA ACE/Server → Configuration Tools → RADIUS Configuration.
2. Select the Authentication Tab.
3. Enable "System Generated PIN's" by checking the selection box
4. Click OK to save these changes.
5. Restart the RSA Authentication Manager to enable the new RADIUS Configuration.

## Mandatory Functionality Testing

1. Authenticate with User 1 using the token and when prompted.
  - Accept the System Generated PIN.
2. Waiting for the Tokencode to change, Authenticate again using the System Generated PIN.

Assuming your authentication was successful; you should see messages in the log monitor similar to the ones in figure 4.



**This completes the Create and Store Node Secret, Force Authentication after New PIN, System-generated PIN, and Name Locking Enabled tests.**

**Note:** If the partner product acts as a RADIUS client, you will see a similar message to Figure 4. The exception being that a Node Secret will not be established.

Date	Time	Current User/Client (Group) Description	Affected User (Site) Server
08/03/2000	15:09:31U	touens/millicity	UPV003376775/Terrell Owen
08/03/2000	11:09:31L	Password Authentication	millicity.securitydynamics
08/03/2000	15:09:32U	touens/millicity	UPV003376775/Terrell Owen
08/03/2000	11:09:32L	PASSCODE Accepted, New PIN Req'd	millicity.securitydynamics
08/03/2000	15:09:32U	touens/millicity	UPV003376775/Terrell Owen
08/03/2000	11:09:32L	Node Secret Sent to Client	millicity.securitydynamics
08/03/2000	15:10:00U	touens/millicity	UPV003376775/Terrell Owen
08/03/2000	11:10:06L	PIN Created by User	millicity.securitydynamics

**Figure 4 – Log Monitor Results**

Date	Time	Current User/Agent Host (Group) Description	Affected User (Site) Server
09/10/2001	20:12:44U	satchue/ps037.securitydynamics.com ---->/	
09/10/2001	16:13:44L	ACCESS DENIED, name lock required ps037.securitydynamics.co	

**Figure 5 – Log Monitor Results**

If the log monitor displays an error similar to the one in figure 5, this means that the RSA Name Lock function has not been implemented correctly.

**Note:** RSA name locking is a mandatory requirement for all certifications integrating Native RSA SecurID Authentication API's. Certification testing cannot continue until this issue has been resolved.

3. Authenticate with User 3 using the token and password.
  - When using the token, enter a user defined 4 Digit PIN.
  - When using the password enter a 4 digit Alphanumeric PIN.



**This completes the User Defined (4-8 Alphanumeric), and 4 Digit Password tests.**

4. Edit system parameters and change PIN from 4-8 to 5-7 and uncheck the box for Alphanumeric PIN Allowed.
5. Edit User 3, clear the user PIN and reset the password.
6. Authenticate with User 3 using the password.
  - Enter a 5 digit Alphanumeric PIN "abcde". You should see Access Denied in the log.
  - Authenticate and enter a 4 digit PIN "1234". You should see Access Denied in the log.
  - Authenticate and enter an 8 digit PIN "12345678". You should see Access Denied in the log.
  - Authenticate again and enter a 5 digit PIN. You should get Passcode Accepted.



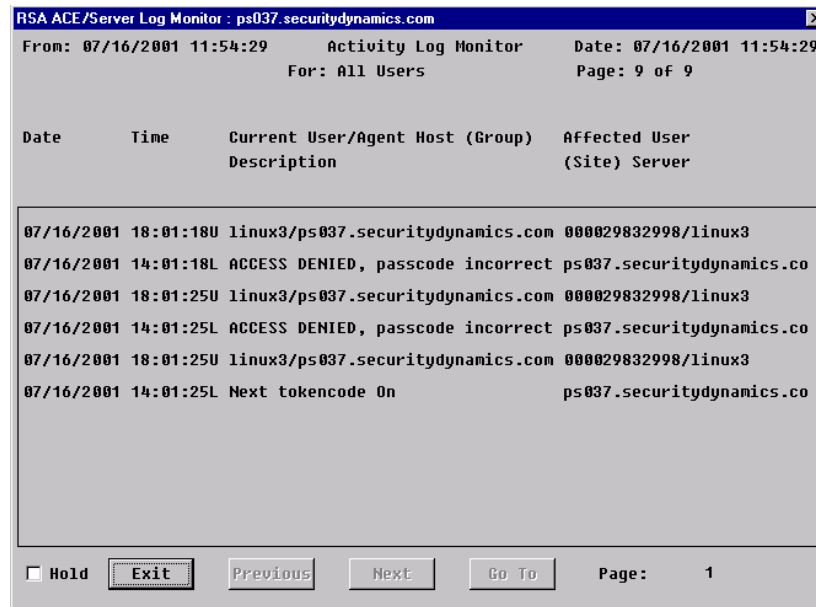
**This completes the User Defined (5-7 Numeric), Deny 4 and 8 Digit PIN and Deny Alphanumeric PIN test.**

7. Authenticate with User 2 using the token and password.
  - When using the token, enter a user defined 6 Digit PIN.
  - When using the password, select a System Generated PIN.



**This completes the User Selectable test.**

8. Authenticate with User 2 and enter an incorrect Passcode three times. This will enable next Tokencode mode for this user.



**Figure 6 – Next Tokencode Mode**

9. Authenticate again and enter a valid Passcode. You should be asked for the Next Tokencode.
10. Enter the Next Tokencode from the token. You should see a message in the log stating Next Tokencode Accepted.




**This step completes the Next Tokencode Mode test.**


RSA ACE/Server Log Monitor : ps037.securitydynamics.com				
From: 07/16/2001 11:54:29		Activity Log Monitor		Date: 07/16/2001 11:54:29
		For: All Users		Page: 9 of 9
Date	Time	Current User/Agent Host (Group)	Affected User (Site) Server	
07/16/2001	18:01:18U	linux3/ps037.securitydynamics.com	000029832998/linux3	
07/16/2001	14:01:18L	ACCESS DENIED, passcode incorrect	ps037.securitydynamics.co	
07/16/2001	18:01:25U	linux3/ps037.securitydynamics.com	000029832998/linux3	
07/16/2001	14:01:25L	ACCESS DENIED, passcode incorrect	ps037.securitydynamics.co	
07/16/2001	18:01:25U	linux3/ps037.securitydynamics.com	000029832998/linux3	
07/16/2001	14:01:25L	Next tokencode 0n	ps037.securitydynamics.co	
07/16/2001	18:02:16U	linux3/ps047.securitydynamics.com	000029832998/linux3	
07/16/2001	14:02:16L	Next tokencode requested	ps035.securitydynamics.co	
07/16/2001	18:03:05U	linux3/ps047.securitydynamics.com	000029832998/linux3	
07/16/2001	14:03:05L	Next tokencode accepted	ps035.securitydynamics.co	

**Figure 7 – Next Tokencode Accepted**

11. Edit system parameters and change PIN back to 4-8 and check the box for Alphanumeric PIN Allowed.
12. Assign the 8-digit token to User 3 and place the user in NEW PIN mode.
13. Authenticate with User 3
  - Enter a PIN of "12345678".

 **This step completes the 16-digit Passcode test**

14. Stop all of the Primary Authentication Manager services.
15. Authenticate with User 2 using the token. (This is part of the Replica testing).
16. Stop all services on one of the Authentication Manager Replicas.
17. Authenticate with User 2 using the token.

 **This step completes the Failover (3-10 Replicas) testing**

18. Stop the second Replica.
19. Authenticate with User 2 using the token. You should get some kind of timeout message.
20. Start the Primary Authentication Manager Master and both Replicas.
21. You should now be able to successfully authenticate with any user.

 **This completes the No RSA Authentication Manager test**

## Additional Functionality Testing

When completing the RSA Secured Certification, the mandatory functionality items must be tested for all partner product integrations. In addition to those requirements, additional testing will be required for certification based on additional functionality that has been included in the partner product.

The RSA Software Token API and Cached Domain Credential handling are two additional features that mandate additional testing be completed for certification. Each of these functions has specific tests designed to verify that the RSA Security API's have been implemented correctly.

In the case of Cached Domain Credential usage, certification testing will also depend on the method in which the password is used, as well as the environment in which it is configured to interoperate.

## Software Token API Testing

If the partner product being tested has integrated the RSA Software Token API, perform the following tests:

1. On the Authentication Manager un-assign all tokens from User 2 and then assign a single Software Token to User 2.
  - Authenticate with User 2 using the Software Token.
  - When prompted accept the System Generated PIN.
  - Authenticate again using the System Generated PIN.
2. Edit User 2, clear the users PIN.
3. Authenticate with User 2 using the Software Token.
  - When prompted, enter the numeric PIN '1234'.
4. Authenticate with User 2 again using the newly created PIN.



**This completes PIN testing for User Selectable**

5. Edit User 2, clear the users PIN and un-check "Required to create a PIN".
6. Authenticate with User 2 using the Software Token.
  - Enter the numeric PIN '12345678'.



**This completes the test for User Defined (8 Digit Numeric)**

7. Edit User 2, clear the users PIN and un-check "Allowed to create a PIN".
8. Authenticate with User 2 using the Software Token.
  - Accept the System Generated PIN.



**This completes the test for System Generated PIN**

9. Authenticate with User 2 and enter an incorrect PIN three times. Enabling Next Tokencode mode for this user.
10. Authenticate again with the Software Token and enter a valid PIN. You should not be asked for the Next Tokencode.

**Note:** Based on the RSA Software Token API functionality, the partner product is expected to automatically calculate the Next Tokencode and then sent this to the RSA Authentication Manager.



**This step completes the Next Tokencode Mode test.**

## Domain Credential Functionality Testing

The RSA SecurID Authentication API version 6.0 offers enhancements that will allow partner products to access cached domain credentials stored within the RSA Authentication Manager database.

This enhancement is intended to allow access to the cached domain credential for very specific purposes. Certification testing will be customized based on the overall solution and how the Cached Domain Credential is actually used by the partner product.

To successfully complete this component of certification testing, a partner product must demonstrate the ability to Determine the State of Cached Credentials on the Authentication Manager; include the ability to Set the Domain Credential and Retrieve the Domain Credential of a user within the RSA Authentication Manager using the provided API.

Additional requirements for certification include but are not limited to the following:

- Interoperate with a Microsoft Active Directory Domain.
- Domain must be protected by RSA Authentication Agent for Microsoft Windows.
- Demonstrate authentication using Cached Domain Credential to a protected domain resource. (e.g. Microsoft GINA, Domain protected Web Listener, etc...)

For more information on certification requirements for this component, contact your assigned Partner Engineering Representative.