RSA SecurID Web Express 1.2
for Windows
Installation and Configuration Guide

RSA SECURITY™

Contact Information

See our Web sites for regional Customer Support telephone and fax numbers.

RSA Security Inc.	**RSA Security Ireland Limited**
www.rsasecurity.com	**www.rsasecurity.ie**

Trademarks

License agreement

Third Party Licenses

This product may include software developed by parties other than RSA Security Inc. To view the text of the license agreements applicable to third-party software in this product, see Help > About in Web Express.

Note on encryption technologies

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

Distribution

Limit distribution of this document to trusted personnel.

RSA notice

The RC5™ Block Encryption Algorithm With Data-Dependent Rotations is protected by U.S. Patent #5,724,428 and #5,835,600.

First printing: April 2003

Contents

Preface

About this Guide

This guide explains how to install and configure RSA SecurID Web Express 1.2.

Getting Support and Service

RSA SecurCare® Online	**www.rsasecurity.com/support/securcare**
Customer Support Information	**www.rsasecurity.com/support**

Before You Call for Customer Support

Note: Customer support is not provided during the warranty period unless a valid Software Service Contract is in force.

Make sure you have direct access to the server running Web Express and the computer running RSA ACE/Server.

Please have the following information about Web Express available when you call:

❑ The name and version of the operating system on which the problem occurs.

❑ The name and version of the web server software running on your web server.

❑ Customization changes that you have made to Web Express.

❑ Any code you have developed that uses the Web Express API calls.

Please have the following information about the RSA ACE/Server available when you call:

❑ Your RSA Security Customer/License ID. You can find this number on the license distribution medium or by running the Configuration Management application on a Windows platform, or by typing ‘sdinfo’ on any UNIX platform.

❑ RSA ACE/Server software version number.

❑ The make and model of the machine on which the problem occurs.

❑ The name and version of the operating system under which the problem occurs.

1 Requirements and Preparations

This chapter lists requirements for installing and using RSA SecurID Web Express. It also explains how to prepare your system before you install the software.

If you want an overview of Web Express, see the *RSA SecurID Web Express 1.2 Planning Guide*.

Software and Hardware Requirements

Components	Supported Software	Minimum Hardware Requirements
Web Server	• Windows 2000 Advanced Server, SP3	• 450MHz or faster processor. • At least 256MB of physical memory. • Server-side Secure Sockets Layer (SSL) is recommended but not required.
RSA ACE/Server	• RSA ACE/Server 5.1	
Web Browser	• Microsoft Internet Explorer 5.5 or later • Netscape Communicator 6.22 or later.	A minimum screen resolution of 1024 by 768 pixels is recommended for using the Manager, Distributor, and Configuration utilities. A minimum screen resolution of 800 by 600 pixels is sufficient for using the end-user utility.

System Requirements

- Web Express and RSA ACE/Server must be installed on separate hosts.
- Because Web Express is critical to the security of your network, RSA Security strongly recommends that you secure the web server that hosts Web Express. For guidelines on securing the
  – Microsoft Internet Information Server (IIS), visit the Microsoft TechNet Security web site at **www.microsoft.com/technet/treeview/default.asp?url=/technet/security/default.asp**.
  – JRun application server, visit the Macromedia Security Zone web site at **http://www.macromedia.com/v1/handlers/index.cfm?ID=23500**.
- Web Express will not function properly, and is not supported, if it is used with an RSA ACE/Server that is out of compliance with license requirements.
- You must stop and restart Web Express every time system PIN parameters are changed on the RSA/ACE Server.

- If you want to use RSA ACE/Server Quick Admin 5.1 with Web Express 1.2, install RSA ACE/Server Quick Admin, then install Web Express by selecting **Web Express Only** as the installation method.

  If you have already installed RSA SecurID Web Express 1.2, uninstall Web Express, install RSA SecurID Quick Admin, and reinstall Web Express by selecting **Web Express Only** as the installation method. For instructions on uninstalling Web Express, see the appendix "Uninstalling RSA SecurID Web Express" in this book.

Preparing to Install

These preparations assume you have already installed these components:

- Web server
- Database to be used by Web Express (either Microsoft Access or another database)
- RSA ACE/Server database

Step 1: Configure RSA ACE/Server to Work With RSA SecurID Web Express

When you install RSA ACE/Server, the services required to use Web Express are installed automatically. However, you must configure the RSA ACE/Server to work with Web Express.

Important: You must configure the Primary RSA ACE/Server to work with Web Express. You cannot use Replicas.

To configure RSA ACE/Server to work with Web Express:

Note: To perform this procedure, you need to know the fully qualified DNS name and IP address of the web server that will host Web Express.

1. Use a text editor to open the **hosts.conf** file.

   This file is in the **ace\prog** directory.
2. Enter the fully qualified DNS name and IP address of the web server that will host Web Express, followed by the hostname and the IP address of the web server.
3. Save and close the file.
4. Use a text editor to open the **sdcommdconfig.txt** file.

   This file is in the **ace\prog** directory.
5. Made sure the **InactivityTimeOut** value is set to 15 minutes.

   This prevents the session on the RSA ACE/Server from timing out before Web Express.
6. Save and close the file.
7. Start the **ACE/Server QuickAdmin Daemon** service.

   For more information, see your RSA ACE/Server documentation.

Step 2: Prepare the RSA ACE/Server Database to Communicate With RSA SecurID Web Express

To prepare the RSA ACE/Server Database to communicate with Web Express:

1. Add a new user to the database and make the user a realm administrator.

   Web Express will use this identity to connect to the RSA ACE/Server.

   Important: You must assign this user an empty RSA ACE/Server task list. If you assign this user tasks, the user identity can be used to affect the RSA ACE/Server using a remote or local administration console.

2. Assign an RSA ACE/Server user password to the new user.

   Important: The password must be static. RSA Security recommends using a password that contains at least eight characters, and includes a combination of both alphabetic and numeric characters.

3. Add the RSA ACE/Server host to the database as an Agent Host. In the Agent Host record, set the Agent Host to be “open” to all locally known users.
4. Add the Web Express server to the database as an Agent Host. In the Agent Host record, set the Agent Host to be “open” to all known users.

For detailed instructions for performing these tasks, see your RSA ACE/Server documentation.

Step 3: Configure the Web Express Database

This section explains how to configure the Microsoft Access database for Web Express. If you are using a different database, see the appendix “Using a Third-Party Database” in this book.

1. Copy these files from the **supported\access_db** directory on the RSA SecurID Web Express 1.2 CD to a directory on the database host, (for example **C:\database**):
   - **RSAArchive.mdb**
   - **RSATransaction.mdb**
2. For each file, right-click the name of the file in Windows Explorer, click **Properties**, and clear the read-only setting.
3. Click **Start** > **Settings** > **Control Panel > Administrative Tools**, and double click **Data Sources (ODBC)**.
4. On the System DSN tab, click **Add**...
5. Select **Microsoft Access Driver [*.mdb]**, and click **Finish**.

   The ODBC Microsoft Access Setup screen opens.
6. Under Database, click **Select**.
7. Locate the **RSATransaction.mdb** file that you copied in step 1 on your computer, select it, and click **OK**.

8. On the ODBC Microsoft Access Setup screen, enter **RSAWE_Transaction_Database** in the **Data Source Name** field, and click **OK**.
9. On the ODBC Data Source Administrator screen, click **OK**.
10. Repeat steps 3 through 5.
11. Locate the **RSAArvhive.mdb** file that you copied in step 1 on your computer, select it, and click **OK**.
12. On the ODBC Microsoft Access Setup screen, enter **RSAWE_Archive_Database** in the **Data Source Name** field, and click **OK**.
13. On the ODBC Data Source Administrator screen, click **OK**.
14. Restart the World Wide Web Publishing Service, and change the startup to Automatic.

Step 4: Collect Required Files

The installation program prompts you for the location of these files:

- **sdti.cer**
- **server.cer**
- **sdconf.rec**

These files are in the **ace\data** directory on your Primary RSA ACE/Server.

Copy all of these files to a single location on the machine that will host Web Express.

Step 5: Collect Required Information

The installation program prompts you for information. Complete the **Value** column in this table, and have the information available when you install Web Express.

Required Information	Value
Where you will install the Web Express software. The default path is **C:\RSAWebExpress**.	
The location of the configuration files you collected in “Step 4: Collect Required Files” on page 12.	
The hostname of the Primary RSA ACE/Server you configured in “Step 1: Configure RSA ACE/Server to Work With RSA SecurID Web Express” on page 10.	
The port number used by the Primary RSA ACE/Server you configured in “Step 1: Configure RSA ACE/Server to Work With RSA SecurID Web Express” on page 10. The default is **5570**.	

Required Information	Value
The User ID for the Web Express account on the RSA ACE/Server. You created this account in "Step 2: Prepare the RSA ACE/Server Database to Communicate With RSA SecurID Web Express" on page 11.	
Password for the Web Express account on the RSA ACE/Server. You created this account in "Step 2: Prepare the RSA ACE/Server Database to Communicate With RSA SecurID Web Express" on page 11.	*Memorize this value. Do not write it down.*
JRun administrator password.	*Memorize this value. Do not write it down.*
Port number for the JRun Management Console. The Web Express default is **8000**.	
Port number for the RSA SecurID Web Express server. The default is **8100**.	
The driver name of the Access (or third-party) database	
The URL to the Access (or third-party) database	
The account number for the Access (or third-party) database administrator	
The password for the Access (or third-party) database administrator	
Whether you will install JRun as a • Service • Application RSA Security recommends installing JRun as a service. For more information, see "Step 1: Install and Set Up RSA SecurID Web Express" on page 15.	

2 Installing and Setting Up RSA SecurID Web Express

This chapter explains how to install and set up RSA SecurID Web Express 1.2 for the first time.

If you want to upgrade Web Express, see the appendix "Upgrading RSA SecurID Web Express" in this book.

Step 1: Install and Set Up RSA SecurID Web Express

There are two types of installations:

Full. Installs this software:

- Java Runtime Environment (JRE) 1.3 - The software that enables JRun to run on Windows.
- JRun 3.1 - The Java application server software.
- RSA SecurID Web Express - The token request, approval, and distribution utilities; web page templates; and the Web Express components that communicate with the RSA ACE/Server.

Web Express Only. Installs Web Express into an existing JRun installation.

There are two ways to run Web Express:

- Service
- Application

If you install Web Express as a service, Web Express starts every time the web server host is booted. If you install Web Express as an application:

- Web Express must be started separately through the Windows Start menu.
- Web Express will stop running if you log out of Windows on the web server.

Note: RSA Security recommends running Web Express as a service.

Important: During set up, you will be prompted to enter the information you collected in "Step 5: Collect Required Information" on page 12. Although this information is required, you can advance through the installation screens even if you do not enter the information. However, if you do not provide the information requested during installation, Web Express will not run.

To install and set up RSA SecurID Web Express:

1. Insert the RSA SecurID Web Express 1.2 CD, navigate to the **\Windows** directory, and double click **setup.exe**.
2. Follow the instructions on your screen, using the information you collected in "Step 5: Collect Required Information" on page 12.

Step 2: Configure the Web Server

Connect the Web Server to RSA SecurID Web Express

To connect the web server to RSA SecurID Web Express:

1. Stop the web server.
2. If the JRun Administration Server is not already started, click **Start** > **Programs** > **RSA Web Express** > **Start JRun Administration Server**.
3. Click **Start** > **Programs** > **RSA Web Express** > **Start JRun Management Console**.
4. At the JRun Management Console login prompt, enter the password for the JRun administrator. You created this password when you installed Web Express.
5. Click **Connector Wizard**.
6. Follow the instructions on your screen, providing information particular to your web server.
7. Close the JRun Management Console.
8. Click **Start** > **Programs** > **RSA Web Express** > **Stop JRun Administration Server**.

Configure SSL on Your Web Server

RSA Security recommends that you configure SSL encryption on your web server. For instructions, see your web server documentation.

Step 3: Enable RSA SecurID Protection on the Administrator Pages

By default, access to the Web Express administrator pages is protected by passwords and RSA SecurID. Because RSA SecurID is more secure than passwords, RSA Security recommends that you use only RSA SecurID authentication, and disable password authentication.

To enable only RSA SecurID protection on the administrator pages:

In the **global_en_US.properties** file, edit the **global_authMethod_admin** tag to read:

```
global_authMethod_admin=SecurID
```

Step 4: Start RSA SecurID Web Express

To start Web Express as a service:

Click **Start** > **Settings** > **Control Panel** > **Administrative Tools** > **Services**, select **RSA Web Express Server**, and click **Start**.

To start Web Express as an application:

Click **Start** > **Programs** > **RSA SecurID Web Express > Start RSA SecurID Web Express**.

3 Configuring RSA SecurID Web Express

Configuring Web Express determines the way Web Express works for Approvers, Distributors, and Configuration Administrators.

This chapter explains ways you can configure Web Express. It also explains tasks related to configuring Web Express (for example, how to start and stop Web Express).

Note: Because Web Express can be configured so extensively, RSA Security recommends that you test your configuration changes in a test environment before enabling them in a production environment.

If you want instructions on how to customize Web Express to meet your company's needs, see the chapter "Customizing RSA SecurID Web Express" in this book.

Starting and Stopping RSA SecurID Web Express

You need to restart Web Express every time you

- Change Web Express templates.
- Change Web Express database connections.

  Note: RSA Security strongly recommends that you change connections and settings by using Web Express. Do not change settings manually. For information, see the Help topic "Edit Connections."

- Change customization data in the Web Express properties files.
- Change system PIN parameters on the RSA ACE/Server.

To start Web Express as a service:

Click **Start** > **Settings** > **Control Panel** > **Administrative Tools** > **Services**, select **RSA Web Express Server**, and click **Start**.

To start Web Express as an application:

Click **Start** > **Programs** > **RSA SecurID Web Express > Start RSA SecurID Web Express**.

To stop Web Express as a service:

Click **Start** > **Settings** > **Control Panel** > **Administrative Tools** > **Services**, select **RSA Web Express**, and click **Stop**.

To stop Web Express as an application:

Click **Start** > **Programs** > **RSA SecurID Web Express > Stop RSA SecurID Web Express**.

Accessing RSA SecurID Web Express

Note: By default, the Web Express pages time out after 15 minutes. If a page times out before you complete a task, when you log back in to Web Express, you must start the task over again. For example, if you complete the Request a Token page and advance to the Confirm Token Request page, but leave Web Express idle for more than 15 minutes, when you log back in to Web Express, you need to start over by completing the Request a Token page again.

User Pages

The user pages enable users to request tokens and perform other tasks. For information about these tasks, see the *RSA SecurID Web Express 1.2 Planning Guide*, and the Help topic "What Web Express Does." To access the Web Express user pages, go to **http://*host_name*/RSASWE/WXUserHome.do**.

Administrator Pages

The administrator pages enable you to perform administrative tasks in Web Express. For information about these tasks, see the *RSA SecurID Web Express 1.2 Planning Guide*, and the Help topic "What Are Web Express Roles." To access the Web Express administrator pages, go to **http://*host_name*/RSASWE/WXAdminHome.do**.

Configuring RSA SecurID Web Express Connections

When Web Express is installed, these connections are specified and encrypted:

RSA ACE/Server. Identifies the RSA ACE/Server with which Web Express communicates and specifies the account on the RSA ACE/Server used to create the connection.

Archive database. Identifies the Archive database and specifies the User ID and password required to access this database. This database stores the requests that are processed through Web Express.

Transaction database. Identifies the Transaction database and specifies the User ID and password required to access this database. This database records events that occur within Web Express.

You must use the Web Express interface to edit these connections. For instructions, see the Help topic “Edit Connections.”

Important: Once these values are encrypted, changing them manually, or changing the **SECURITY_BLOCK_DO_NOT_REMOVE** setting in the **config.properties** file will cause the encryption to fail and Web Express to stop working. Change these connections only through Web Express. For information, see the Help topic “Edit Connections.”

Managing Administrators and Cost Centers

You create Web Express administrators by assigning people these roles:

- Approver - Reviews and approves token requests
- Distributor - Reviews and deploys approved token requests
- Configuration Administrator - Configures and customizes Web Express

You can distribute workloads among Approvers and Distributors by creating organizational units called cost centers.

For detailed information about creating and managing administrators and cost centers, see the Help topics “Managing Administrators” and “Managing Cost Centers.”

Configuring the Approval Process

You can configure the Approval process in these ways:

- Specify whether the Approver role is enabled or disabled.
- Specify whether all Approvers will, by default, automatically receive e-mail notification when requests enter their queues.
- Require Approvers to log in using RSA SecurID.

To configure the Approval process, use the Web Express administrator pages. For more information, see the Help topic “Configure the Approver Role.”

To require Approvers to log in using RSA SecurID, see “Step 3: Enable RSA SecurID Protection on the Administrator Pages” on page 16.

Configuring the Distribution Process

You can configure the Distribution process is these ways:

- Specify whether the Distributor role is enabled or disabled.
- Specify whether all Distributors will, by default, automatically receive e-mail notification when requests enter their queues.
- Require Distributors to log in using RSA SecurID

To configure the distribution process, use the Web Express administrator pages. For more information, see the Help topic "Configure the Distributor Role."

To require Distributors to log in using RSA SecurID, see "Step 3: Enable RSA SecurID Protection on the Administrator Pages" on page 16.

Configuring RSA SecurID Web Express e-Mail

You can configure Web Express e-mail in these ways:

- Specify the mail server connection, and the User ID and password required to access the mail server
- Enable automatic e-mail notification to notify:
  - Requesters about the status of their requests
  - Approvers and Distributors when requests enter their queues
  - Third-party distribution houses when requests are deployed

  Enabling automatic e-mail includes specifying:

  - Templates used for the notifications
  - Subject and address lines of the e-mails
  - The URL that requesters visit to activate software tokens

To configure the e-mail settings, use the Web Express administrator pages. For instructions, see the Help topic "Configure RSA SecurID Web Express e-Mail."

Loading Token Requests in Bulk

Instead of requiring requesters to submit token requests individually, you can load requests into Web Express in bulk. The next procedure tells you how to do this. The specific steps vary depending on the database you are using (either Microsoft Access, or a third-party database).

To load requests in bulk:

1. Submit a request for a new token in Web Express, but do not process the request.

   This request sets the order of the request information in the request database. In the next step, you use the request database to create a template that will contain the bulk requests.
2. Open the Web Express request database (**RSATransaction.mdb**), and export the contents into a comma separated value (CSV) file.

   For instructions, see the documentation provided with your particular database. This creates a template that will contain the bulk requests.
3. Create a CSV file of the bulk requests, and order the information according to the template you created in the previous step.
4. Import the CSV file containing the requests into the Web Express request database.

Starting the RSA ACE/Server Quick Admin Daemon

The RSA ACE/Server Quick Admin daemon is a back-end service that Web Express uses to communicate with the RSA ACE/Server. The service runs on your Primary RSA ACE/Server and starts automatically when the RSA ACE/Server is started.

If the RSA ACE/Server Quick Admin daemon is stopped, Web Express tasks that require communication with the RSA ACE/Server fail. For example, if a requester tries to change a PIN, the requester sees an error message with instructions to contact the Web Express administrator.

Note: This procedure assumes you are using an RSA ACE/Server for Windows.

To start the RSA ACE/Server Quick Admin daemon:

On the Primary RSA ACE/Server host, click **Start** > **Settings** > **Control Panel** > **Services** > **RSA ACE/Server QuickAdmin Daemon** > **Start**.

If the service fails to start, see “Problems Communicating With the RSA ACE/Server” on page 52.

Enabling the RSA SecurID Web Express APIs

Web Express ships with a set of customizable APIs. For information about customizing the APIs, see the appendix "Using the RSA SecurID Web Express APIs" in this book. To enable one or more of the APIs, use the Web Express interface. For instructions, see the Help topic "Enable RSA SecurID Web Express APIs."

4 Customizing RSA SecurID Web Express

This chapter explains how to customize the way Web Express works for users, and what appears on the user pages.

The Web Express interface depends on a set of properties files stored on the Web Express host web server. The properties files are text files that control the display of the Web Express pages. Using a text editor, you can edit the files to customize the appearance and functionality of the pages.

This chapter describes the format of the properties files. It also contains procedures for performing common customization tasks. For example, you might want to restrict the type of token that users can request (see "Removing Token Types From The Drop-down List" on page 32).

Before you begin customizing the properties files, be aware of the following:

- All properties files are stored in language-specific directories in the **RSAWebExpress/JRun/servers/default/rsaswe/WEB-INF/classes/i18n** directory.

  For ease of use and localization, the names of the properties files in this chapter do not include the country code or file extension. For example, instances of **WXRequestToken** in this chapter refer to the file **WXRequestToken_en_US.properties** on a server configured to use English for the United States. For more information on localizing Web Express, see "Localizing Web Express"on page 29.
- RSA Security strongly recommends that you back up all of the properties files before you make any changes to them.
- After editing a properties file, you must restart Web Express for the changes to take effect. For instructions, see "Starting and Stopping RSA SecurID Web Express"on page 19.

If you want instructions on how to customize the way Web Express works for Web Express administrators, see the chapter "Configuring RSA SecurID Web Express" in this book.

Important: After editing a properties file, you must restart Web Express for the changes to take effect. For instructions, see "Starting and Stopping RSA SecurID Web Express"on page 19.

Format of the Properties Files

Each properties file controls the display of one or, in some cases, two Web Express pages. The files contain a list of variables (called “attributes”) and values for each element, including

- page title
- introductory text
- buttons and button names
- fields and field names

In most cases, you do not edit the attribute itself; just the value. If you do not want certain fields, buttons or text to display in the page, you can set the **_visible** attribute to **false**.

The files are divided into named sections and include comments designed to assist you in customizing the Web Express end user experience to your particular needs. See “Sections of the Properties Files” on page 27 for a description.

Attributes

Attribute names contain three parts:

- prefix
- name
- suffix

Attribute names take the following form in the properties file:

```
prefixName_suffix
```

The list of standard prefix and suffix values are described in the **wrap** properties file in the **RSAWebExpress\JRun\servers\default\rsaswe\WEB-INF\classes\i18n\en_US\** directory. The name states the purpose or functionality of the particular field in the page. For example, open the **WXUserRequestToken** properties file, which controls the display of the Request Token page. The first, unlabeled section of the file contains comments (including the file name and the name of the page in Web Express that is generated from the file) and the general attributes. Go to the section labeled “Token Information.” Look at the first attribute **ghTokenInfo_label**.

The prefix is **gh** and the suffix is **_label**. These are examples of some of the standard prefix and suffix conventions used in Web Express. The **TokenInfo** part of the name is based on the functionality of the section of the Token request page in which the attribute occurs, and becomes the key name for the other attributes in the Token Information section of the Request Token page.

Whenever you need to add attributes to any of the properties files, the procedures in this guide provide the exact prefix, name and suffix for the attribute.

Values

Some attributes have a restricted set of possible values, while others are restricted only by the characters or numbers that you can use when you set the values. Comments in the properties files provide details on any relevant restrictions. For example, open the **WXUserRequestToken** properties file, and go to the section labeled CUSTOM FIELDS. The following text explains the function and the possible values of the **_parameter0** attribute:

```
# _parameter0=  specifies whether or not the input is
# processed by the Web Express API.
# Possible values are:
# API_INACTIVE - which does not process the input through
#               the Web Express API
# API_CUSTOM# - where # is a number between 1 and 16
```

Sections of the Properties Files

There are comments throughout the properties file that are designed to help you while you edit the files. The properties files are separated into named sections, with comments describing each section. Some sections, such as those that specify button names and images, do not need to be edited at all, so the comments specifically state **DO NOT EDIT**.

Introductory Comments

This section of the properties file contains the filename and a brief description of the purpose of the file.

General attributes

Most of the properties files contain one or more of these general attributes:

autogenerate

The first attribute in each properties file is the auto-generate attribute,

```
autogenerate=start
```

This is a required attribute that Java uses to generate the page.

title

This attribute controls the title of the page:

```
title=
```

If the properties file controls more than one page, there may be more than one “title” attribute in a properties file. For example, the Request Token page and the Confirm Request Token page share the same properties file (**WXUserRequestToken)**, so the following two attributes appear in the file:

```
title=Request Token
titleConfirm=Confirm Your Token Request
```

helpPaneVisible

This attribute controls the Help panel that displays on the right side of the browser window:

```
helpPaneVisible=
```

Possible values are **true** or **false**. If you set the value to true, and the page contains no default Help file, you must create a Help file in HTML format and specify its location so that it will display in the Help panel when the page displays. Most of the end user pages contain no default Help files. The Administration pages, including those pages accessible by Approvers and Distributors do contain Help.

requiredIndicatorField

This attribute displays a reminder at the top of the page that users must enter values in the fields marked with an asterisk (*). Set this value to **true** when there are required fields on the page.

authMethods

This attribute, when set, protects pages with the Web Express Password, RSA SecurID or RSA Question and Answer method of authentication. For more information about this attribute, see "Specifying Authentication Methods on a Page" on page 46.

authForce

This attribute, when set to **true**, requires a user or administrator to authenticate every time they access a protected page. The normal behavior of Web Express is to cache authentication status on a per session basis, so that only one successful authentication is required to access protected pages.

introText

This attribute contains the text that displays on the page, directly under the title of the page. RSA Security ships Web Express with default introductory text for each page, but some of the pages contain introductory text that will be more informative to your users if you edit it to fit your system. For example, the Request Submitted page that displays when a user has successfully submitted a request for a token contains the following message in the **introText** attribute:

```
Your token request has been received. You will be notified
about the status of your request.
```

This message is intentionally ambiguous, so that you can edit it to include information about when and how a user will be notified of the status of the token request.

There may be more than one **introText** attribute in a properties file if the file controls more than on page. For example, the Request Token page and the Confirm Request Token page share the same properties file (**WXUserRequestToken**), so the following two attributes appear in the file:

```
introText=
introTextConfirm=
```

Closely related to the **introText** attribute is the **headline** attribute, which is used to emphasize text.

Localizing Web Express

Web Express supports any of the languages supported by your browser. By default, Web Express includes properties files configured to use the en-us (English for the United States) language code. For example, the Request Token page uses the **WXUserRequestToken_en_US.properties** file. If you want to change the language to Chinese for Hong Kong, you must use the zh-hk language code. In this case, the properties file for the Request Token page would be named **WXUserRequestToken_zh_hk.properties**.

To use localized versions of Web Express pages:

1. Create a new directory in the **RSAWebExpress\JRun\servers\default\rsaswe\WEB-INF\classes\i18n\** directory.

   Name the directory using the two or four letter language code of the local country. For example, the directory for English for the United States is **en_US**. The directory for Chinese for Hong Kong would be **zh_hk**.

2. Copy the default properties files from the **en_US** directory to the new directory.

When a user accesses Web Express pages with a browser that is configured to use the language you specify, Web Express displays localized pages. After you create localized versions of the pages, if you want to perform any additional customization, you must edit each version of the properties file.

Customizing the Token Request Pages

To request tokens, users must access the Request a New Token page. This page contains a set of default fields, configurable fields and up to 16 custom fields. The token request feature of Web Express is on by default, but you can disable it, allowing users to change PINs only. For more information, see "Enabling and Disabling the Token Request Feature"on page 31.

Default Fields

The following fields display on the Request Token page by default:

- **Token Type**.This is a drop-down list of the types of tokens available. By default, the list contains these token types:
  - PC Software Token
  - Pocket PC Software Token
  - Key Fob
  - Standard Card
  - PINPad

  In Web Express, the key fob, standard card and PINPad are hardware tokens.

  You can customize the list to include only those token types that your company uses, or add other token types. For more information, see "Removing Token Types From The Drop-down List" on page 32 and "Adding a New Token Type" on page 33.
- **User ID**. This is the login name the requester uses to log in to the network or access protected resources.
- **First Name** and **Last Name**.
- **E-mail Address**. This is the address of the requester's e-mail account. Web Express sends the approval e-mail message to this address. If the token requested is a software token and Web Express is configured to deliver software tokens by e-mail, the message includes the software token file. For more information, see "Customizing Approval and Rejection e-Mail Messages" on page 36 and "Software Token Delivery" on page 38.
- **Cost Center**. The department in your company that is charged for the cost of the token. If requesters do not know their cost centers, include the information in the approval e-mail.
- **ACE/Server Group**. This field allows requesters to assign themselves to a group in the ACE/Server database. For more information, see "Specifying RSA ACE/Server Groups" on page 35.

Configurable Fields

- **Activation Password** and **Confirm Password**. These fields provide additional security to ensure that the user activating the token is the same user who requested it. Requesters enter a one-time password that only they know. Web Express prompts requesters to enter the same password when they activate their tokens. For more information, see "Enabling and Disabling the Activation Password" on page 34.

The following fields do not display by default, but are pre-configured in the properties file (**WXUserRequestToken**) for the Request Token page. You can edit the properties file to display these fields.

- **Token Name**. This field allows the requester to specify a name for a PC Software Token or Pocket PC Software Token. The field displays only when the requester selects one of the software token types from the Token Type list. The Token Name is a useful way to distinguish between multiple software tokens.
- **Pocket PC Serial Number**. This field requires requesters to enter the serial number of the handheld device on which the requested Pocket PC Software Token will be installed. Requiring the serial number restricts installation of the token to the specified device.

Custom Fields

For each token type, there is a set of attributes in the CUSTOM TOKEN and CUSTOM USER sections of **WXUserRequestToken** file. These attributes define the custom fields that display when a user selects a token type. The custom fields allow you to gather additional information from users when they request tokens. For example, when the Pocket PC type is selected, an additional field for specifying a Device Nickname could display in the Token Information section of the Request Token page.

Enabling and Disabling the Token Request Feature

You can configure Web Express to not allow users to request tokens, but still allow users to change their PINs.

To disable token requests:

1. On the Admin Home page, click **Configure System** > **System Setup** > **System Settings**.
2. Under **Web Express Features**, clear the **Token Request management** checkbox.

   As stated on the System Settings page, if you disable token request management, user account management must be enabled.
3. At the bottom of the page, click **Save**.

   Web Express allows users to change PINs, but not request tokens.

Configuring the Drop-down List

Each token type is defined by the following two attributes in the **WXUserRequestToken** properties file:

- **tsTokenInfoSelect_option#**, which defines the text that displays in the drop-down list.
- **tsTokenInfoSelect_optionValue#**, which defines an internal value used by Web Express to identify the token type.

For each token type, you can configure 16 custom fields: 8 fields of token information and 8 fields of user information. By default, the information collected in custom fields is stored in the Web Express database. Additionally, the information can be stored in the RSA ACE/Server database in token or user extension fields. Web Express can also send the information in these fields to the Web Express API. For more information, see the appendix "Using the RSA SecurID Web Express APIs" in this book and the comments in the **WXUserRequestToken** properties file.

Each custom field has a set of 8 attributes. The attributes have either one of these prefixes:

- **vtTokenInfo**. Define custom fields in the Token Information section.
- **vtUserInfo**. Define custom fields in the User Information section.

The names of the attributes start with these prefixes. The suffix of the name must contain the following:

- A designation of the token type (either **Hardware** or some other designation that indicates a software token type)
- A number (used to identify the custom field)
- A standard suffix

The suffixes and the possible values of the attributes are listed in the section CUSTOM FIELDS in the **WXUserRequestToken** properties file.

Removing Token Types From The Drop-down List

You can remove token types that your company does not use by commenting out the lines in the **WXUserRequestToken** properties file that contain the attributes that define the entry in the list.

To remove a token type from the drop-down list:

1. Open the file **WXUserRequestToken**.
2. Go to the section TOKEN TYPE DROPDOWN LIST ATTRIBUTES.
3. Comment out the following two attributes for the token type that you want to remove:

   ```
   tsTokenInfoSelect_option#=token type
   tsTokenInfoSelect_optionValue#=token type
   ```

   where # is the number that specifies the token type that you want to remove.

   For example, to remove Standard card from the list of token types, comment out the following two lines:

   ```
   tsTokenInfoSelect_option3=Standard card
   tsTokenInfoSelect_optionValue3=HardwareB
   ```

4. Edit any attributes following the attribute you comment out so that the numbers used to specify the attributes are consecutive numbers.

   For example, if you removed standard card from the list, as described in the example in step 3, you must change the attributes for the PINPad type to use the number 3:

   ```
   tsTokenInfoSelect_option3=PINPad
   tsTokenInfoSelect_optionValue3=HardwareC
   ```

   The numbers you use in the attribute names must be consecutive, with the first set of attributes using the number 0.

Adding a New Token Type

Creating new token types enables you to configure the Request Token page to display different custom fields for different types of users who may be requesting the same type of token, but whom you may want to require to input different information in the custom fields. For example, if you give tokens to employees and non-employees (such as business partners or customers), you may want to display different custom fields for each type of user.

To create a new token type, add an additional attribute-value pair (as described in "Configuring the Drop-down List" on page 31) to the list, incrementing the attribute number by one. The value of the **_option** attribute must also be used as part of the name of the attributes you add to the CUSTOM TOKEN and CUSTOM USER sections.

To add a new token type to the drop-down list:

1. Open the **WXUserRequestToken** properties file.
2. Under TOKEN TYPE DROPDOWN LIST ATTRIBUTES, after the pair of attributes that define the last token type in the list, add these attributes:

   ```
   tsTokenInfoSelect_option#=
   tsTokenInfoSelect_optionValue#=
   ```

   where

   - # is an integer that is one greater than the number specifying the last token type in the section.

     The numbers you use in the attribute names must be consecutive, with the first set of attributes using the number 0.
   - the **_option** attribute is the text that displays in the drop-down list on the page.
   - the **_optionValue** attribute is an internal value used by Web Express to identify the token type.

   Note: When you define the custom user and custom token fields for the new token type, you must use the value of the **_optionValue** attribute you define here as part of the name of the attributes that define the custom fields. The **_optionValue** for any hardware token type must include the word **Hardware**.

3. In the CUSTOM FIELDS section, after the set of attributes that define the custom token fields for the last token type in the list, add a set of attributes that have the following suffixes and define values for them for each custom token field you want to display when the requester selects the new token type from the drop-down list.

   ```
   _label=Future Defined
   _visible=false
   _minLimit=0
   _maxLimit=80
   _hint=
   _parameter0=API_INACTIVE
   _parameter1=NOT_EXTENSION
   _parameter2=FUTURE_DEFINED
   ```

   The complete name of the attribute must have the prefix **vt** followed by **TokenInfo** or **UserInfo**, followed by the value of the **_optionValue** attribute you defined in the previous step.

   For example, if you added a token type for vendor's requesting software tokens, you might add the following two attributes to the drop-down list:

   ```
   tsTokenInfoSelect_option5=Software Token for Vendors
   tsTokenInfoSelect_optionValue5=SoftVendor
   ```

   In this case, the following attributes would specify the first custom token field for vendor software tokens:

   ```
   vtTokenInfoSoftVendor1_label=
   vtTokenInfoSoftVendor1_visible=
   vtTokenInfoSoftVendor1_minLimit=
   vtTokenInfoSoftVendor1_maxLimit=
   vtTokenInfoSoftVendor1_hint=
   vtTokenInfoSoftVendor1_parameter0=
   vtTokenInfoSoftVendor1_parameter1=
   vtTokenInfoSoftVendor1_parameter2=
   ```

4. In the CUSTOM FIELDS section, after the set of attributes that define the custom user fields for the last token type in the list, add the same set of attributes as you added in step 3.

5. Save and close the file.

Enabling and Disabling the Activation Password

The activation password provides an additional level of security for verifying that the recipient of the activation code is indeed the requester of the token. On the Token Request page, the requester enters a password in the **Activation Password** field. The requester must provide the same password when completing and submitting the Activate Token page.

To require an activation password:

1. On the Admin Home page, click **Configure System** > **System Setup** > **Workflow & User Account**.
2. Under Request & Approval Process, from the **Automatic Approval** drop-down list, select **Enabled. Allow requests by existing ACE/Server users**.
3. At the bottom of the page, click **Save**.

Specifying RSA ACE/Server Groups

You can configure Web Express to enable requesters to assign themselves to a group in the RSA ACE/Server database. The Request Token page contains the ACE Group field by default, but using this capability requires some administration on the RSA ACE/Server itself. You must have administrator privileges on the RSA ACE/Server to complete this task.

To specify ACE groups in Web Express:

1. Create a group in the RSA ACE/Server database.
    - On the RSA ACE/Server, click **Start** > **Programs** > **RSA ACE/Server** > **Database Administration - Host Mode**.
    - In the Database Administration application, click **Group** > **Add Group**.
    - In the Add Group dialog box, click **Help** for further instructions.
2. Open the **WXUserRequestToken** properties file.
3. Under RSA ACE/SERVER GROUPS, for each group that you want to display in the ACE Group list, add the following two attributes:

    ```
    tsUserInfogroupAccess_option#=
    tsUserInfogroupAccess_optionValue#=
    ```

    where

    - # is an integer that is one greater than the number specifying the last group in the list.

      The numbers you use in attribute names must be consecutive, with the first set of attributes using the number 0.
    - the **_option** attribute is the text that displays as the name of the group in the ACE Group drop-down list.
    - the **_optionValue** attribute is the name of the group as it is defined in the RSA ACE/Server database.
4. Save and close the **WXUserRequestToken** properties file.
5. Open the **WXAdminEditAppReqDetails** properties file.
6. Under RSA ACE/SERVER GROUP FIELDS, for each group that you want to display in the ACE Group list, add the same attributes and values that you added in step 3.

    The **_optionValue#** attribute defines the text that displays as the group name in the details.

7. Save and close the **WXAdminEditAppReqDetails** properties file.
8. Remember to tell your users the name of the group they need to select when requesting a token.

Allowing Users to Request Additional Tokens

By default, Web Express does not allow a request to be automatically approved if the requester already has a user record in the RSA ACE/Server database. If you have any requesters who require more than one token, or who already have a token assigned to them, and you want to use auto-approval, you must configure Web Express to allow these users to request additional tokens.

To allow users to request additional tokens:

1. On the Admin Home page, click **Configure System** > **System Setup** > **Workflow & User Account**.
2. Under Request & Approval Process, for Automatic Approval, select **Enable. Allow requests by existing ACE/Server users**.
3. At the bottom of the page, click **Save**.

Web Express can now process requests from existing users when automatic approval is enabled.

Customizing Approval and Rejection e-Mail Messages

If you configure Web Express to send e-mail notification of the status of requests to requesters, you can change the content of the e-mail by editing these text files in the **\RSAWebExpress\JRun\servers\default\rsaswe\WEB-INF\templates\language\en** directory:

- **UserConfirmMessage.txt** - Template for the body text of the approval e-mail
- **UserRejectMessage.txt** - Template for the body text of the rejection e-mail

Note: This section describes only how to edit the text of the e-mail messages. To configure Web Express to send the e-mail, you must configure the Request Approval and Request Rejection sections of the E-mail administration page. For more information, see the Help topic, “Customize Web Express for Users.”

Approval e-Mail

The approval e-mail contains the requester’s user ID and an activation code. By default, the e-mail presents this information in two ways: in a URL that links to the Activate Token page, and in plain text. If your e-mail system allows active links within e-mail messages, using the first method automatically fills the **User ID** and **Activation Code** fields on the Activate Token page. If you do not want to use the active link, requesters must access the Activate Token page from the Web Express User Home page and enter the information manually.

The approval e-mail template (**UserConfirmMessage.txt**) contains both the active link and the static text. You can configure the e-mail to use either or both of these formats. You can use any standard text editor to edit the template.

The approval e-mail can contain a software token file when the requested token is a software token and the software token delivery method is e-mail. For more information, see "Software Token Delivery" on page 38.

Customizing e-Mail for Downloading Software Tokens

The default wording in the template applies to hardware tokens and software tokens delivered by e-mail or some form of physical distribution. If you have configured Web Express to allow requesters to download software tokens, you must change the text of the e-mail to indicate that they can begin the activation process when they receive the approval e-mail, not when they receive their tokens. When the token delivery method is download, the requester receives the token as part of the token activation process.

Description of the Variables in the Template

The approval e-mail template contains variables with values that specify information specific to the requester and the request. This table lists the variables and where their values originate:

Variable	Value
FIRST_NAME, USERID	Provided by the requester on the Request Token page.
APPROVAL_LINK	Configured by the Web Express administrator as the Activation URL under Request Approval on the E-mail page.
APPROVAL_CODE	Randomly generated by Web Express for each request.

Rejection e-Mail

The rejection e-mail contains this information about the requester:

- First and last names
- User ID
- e-Mail address
- Cost center
- Date of the rejection
- Reason for the rejection in the form of a rejection code

You can edit the rejection e-mail template (**UserRejectMessage.txt**) to delete any information that you do not want to communicate to users. Additionally, you may want to edit the contact information to include a specific administrator and the e-mail address of the administrator.

Software Token Delivery

You must specify the method that requesters use to get their software tokens. There are three software token delivery methods, configured in the Workflow & User Account page, under **Token Distribution**, in the **Software Token Distribution** drop-down list:

- **No Distributor. Requesters download tokens**. Allows requesters to download tokens from the Token Download page. Web Express installs with this method as the default.
- **No Distributor. E-mail tokens to Requesters**. Delivers the software token file as an attachment to the approval e-mail.
- **Distributor manually delivers tokens**. Requires the Distributor to copy the software token file from the token repository to a physical medium (such as a diskette), deliver it to the requester manually, and mark the token as deployed in Distributor queue.

In all cases, the software token application must be installed on the requester's PC or handheld device in order to install and use the software token. For more information, see "Downloading the Software Token Application" on page 39 and the Help topic, "Customize Web Express for Users."

Download

When you configure Web Express to use the download method of software token delivery, the Token Download page displays immediately after the requester activates the token. The requester must download and save the software token file to the hard drive, and then install it according to the software token application Help.

As a convenience, the Token Download page allows the requester to download the platform-specific software token application required to use the token. For more information, see "Downloading the Software Token Application" on page 39.

e-Mail

When you configure Web Express to use the e-mail method, the software token is sent as an attachment to the approval e-mail. You can edit the text of the approval e-mail to include some information about the software token. For more information, see "Approval e-Mail" on page 36.

Manual

The manual method requires the most administrator involvement. The Distributor must copy the software token file to a diskette or other physical medium and deliver the medium to the requester.

Downloading the Software Token Application

To use software tokens, users must have the software token application installed on their PCs or handheld devices. The Token Activated page enables users to download the software token application they need. Currently, Web Express supports software tokens for PC and Pocket PC.

If you choose to pre-install the token application for users, or assign the task of installation to administrators, you can remove the link that enables users to download the token application or you must tell users that they do not need to download the application.

Downloading the Token Application from Your Own Server

By default, the Token Download Complete page (**WXUserThankYou** properties file) links to a Download page on the RSA Security web site that allows users to download the software token application. This Download page requires users to enter personal information before downloading the application. To simplify the process, RSA Security also provides the token applications on the Web Express CD, in the **supported\token_apps** directory, so that you can post the application files on your own web server. Users can download the application without providing any personal information to RSA Security.

If you choose to store the application files on your own server, you must edit the **WXUserThankYou** properties file and the User Home page so that they link to the location of the application files. This properties file contains additional text in the introText attribute that is commented out by default, but which you can uncomment and use to specify the correct links to your web server. The User Home page link is in an HTML file (**home_tutorial.htm**) in the **RSAWebExpress\JRun\servers\default\rsaswe\html** directory.

Important: If you choose to configure Web Express to access the files on your own server, remember to keep the copies up to date by downloading the latest versions from the RSA Security web site at **http://www.rsasecurity.com/go/webexpress**.

To configure downloading the token application from your own web server:

1. Create a directory on your web server to store the token application files.
2. On the Web Express CD, go to the **supported\token_apps** directory and copy the the **.zip** files to the directory you created in step 1.
3. On the Web Express server, in the **\RSAWebExpress\JRun\servers\default\rsaswe\WEB-INF\classes\i18n\en_US** directory, open the **WXUserThankYou** properties file.

4. Comment out the following lines of the value of the introText attribute:

   ```
   <a
   href="http://www.rsasecurity.com/go/webexpress">Download
   Software Token \
   Application</a> to download the application from the RSA
   Security Corporate \
   Web site, and then install the application.
   ```

5. Uncomment the following lines:

   ```
   #one of the following links to download the application
   for your platform \
   #and then install it.
   ```

6. Uncomment any of the following lines that apply to your system.

   ```
   # <li><a href=" ">Desktop PC</a> \
   # <li><a href=" ">Pocket PC</a> \
   ```

   For each line that you uncomment, specify the location of the token application zip file for that platform in the space between the quotation marks.

7. If you specify two or more platforms, and want the links to the token application zip files to appear in a bulleted list, uncomment the following lines:

   ```
   # <ul>
   ```

   and

   ```
   # </ul>
   ```

8. Save and close the **WXUserThankYou** properties file.

9. On the Web Express server, in the **RSAWebExpress\JRun\servers\default\rsaswe\html** directory, open the **home_tutorial.htm** file in a text editor.

10. Find the following text in the **home_tutorial.htm** file:

    ```
    <tr>
    <td width="8" valign="top"><img src="images/caret_gray.gif"
    width="10" height="10" valign="top"><br>&nbsp;<br>&nbsp;
    </td>
    <td width="100%"  valign="top"> <a
    href="JavaScript:popup(1,'http://www.rsasecurity.com/go/webe
    xpress/',0)"  class="grp_hdr"> &nbsp;Download Software Token
    Application</a> from RSA Security Corporate Web site. This
    will open a new browser window.
    ```

11. Edit the text of **<a href** tag to include only a link to the application file on your Web Express server. For example,

    ```
    <a
    href=”http://your_server_name/application_directory/app.zip”
    >
    ```

12. Delete the following text:

    ```
    from RSA Security Corporate Web site. This will open a new
    browser window.
    ```

13. Add text that describes the token type. For example,

    ```
    for Pocket PC.
    ```

14. For each software token type that you want to allow requesters to download, copy the edited text, paste it in the file after the edited text, change the link to point to the location of the additional token type and edit the descriptive text.

15. Save and close the **home_tutorial.htm** file.

Setting Up Question & Answer Authentication

Question and Answer (Q & A) authentication allows users to authenticate without an RSA SecurID token. This method is useful when a user forgets the PIN of his or her token and needs to access the Change PIN page, which, by default, is configured to allow users to select between two methods: RSA SecurID or Q&A authentication.

The Q & A method prompts users to provide answers to a set of previously answered questions and authenticates the user (or not) based upon the number of correct answers the user provides. Before users can successfully authenticate using the Q & A method, they must set up their own questions and answers through the Q & A Setup page. As the administrator, you can use the default list of questions provided with Web Express or create a list of your own questions. In either case, users choose a subset of these questions and provide answers that they feel they can easily remember.

By default, answers are stored in the RSA ACE/Server database, in user extension fields in the user record. When a user accesses a page that is protected by the Q & A method, the user's input is verified against the answers in the RSA ACE/Server database. However, Web Express provides an API that allows the input to be verified against data stored in a third-party database or an LDAP directory. For more information, see "Using the API to Verify Answers" on page 44.

To use the Q & A authentication method, you must perform the following tasks:

- Specify the Q & A authentication method as an allowed method of authentication on pages you want to protect.

  For more information, see "Specifying Authentication Methods on a Page" on page 46.

- Configure the number of questions that will be asked.

  By default, users are asked five questions.

- Configure the number of correct answers that must be provided in order to authenticate successfully.

  By default, users must answer three questions correctly.

- Configure the list of selectable questions in the Set Up Q & A Authentication page.

  Web Express provides 12 default questions. Your users will choose the questions they want to answer from this list when they access the Set Up Q & A Authentication page.

- Notify your users that they must select questions and provide answers through the Set Up Q & A Authentication page.

Configuring the Number of Questions

You must specify the number of questions that users must answer and the number of questions that users must answer correctly (also known as the Question Threshold) in order to authenticate successfully. RSA SecurID Web Express provides a list of default questions in the **AuthQARegistration** properties file.

To configure the number of questions and correct answers:

1. In the Web Express Admin Home page, click **Configure System** > **System Setup** > **Workflow & User Account**.
2. Under **Question and Answer Page Settings**, in the **Number of Questions** field, enter the number of questions that you want the user to answer.
3. In the **Question Threshold** field, enter the number of questions that the user must answer correctly.
4. Click **Save** to save the settings.

Configuring the List of Questions

RSA SecurID Web Express provides a list of default questions in the **AuthQARegistration** properties file. You can use the default questions or create your own questions. The easiest way to use your own set of questions is to edit the default list, replacing the default questions with your own questions, adding additional attributes for any additional questions you want to display in the list, and commenting out the attributes of any extra questions.

Additionally, during an authentication, the Web Express API can validate answers against a third-party data source, such as an LDAP directory, instead of the RSA ACE/Server database.

Important: If you configure a question to be validated using the API, be aware that Web Express does not store the answer provided by the user in the third-party data source. For more information, see "Using the API to Verify Answers" on page 44.

To configure the list of questions using the default list:

1. Open the **WXAuthQARegistration** properties file.
2. Under DEFAULT QUESTION LIST, change the value of one of the **defaultQuestionList#** attributes to the text of your question.

   To add a question to the existing list, add the following attribute after the last attribute in the list:

   ```
   defaultQuestionList#=question
   ```

   where # is one greater than the number used in the last attribute in the list and *question* is the text of the question.
3. Save and close the file.

Configuring the default list in this way does not require users to answer particular questions. Users are allowed to choose from all the available questions when they set up Q & A authentication. You can configure the Set Up Q & A Authentication page so that a question does not have a selectable list, but instead displays only one static question.

To configure a question so that all users must set up an answer:

1. Open the **WXAuthQARegistration** properties file.
2. Decide which one of the questions you want to be a static question.

   The examples in this procedure use the **defaultQuestionList1** attribute.
3. Under FIELD ATTRIBUTES, find the set of attributes that uses the same number as the attribute you chose in step 2. For example,

   ```
   tsQuestion1_label
   tsQuesion1_useDefaultList
   ```
4. Change the value of the **tsQuestion1_useDefaultList** attribute to **NO. For example,**

   ```
   tsQuesion1_useDefaultList=NO
   ```
5. Add the attributes **tsQuestion#_option0** and **tsQuestion#_optionValue0**, and set the value to the text of the question that must be answered. For example,

   ```
   tsQuestion1_option0=What is your e-mail password?
   tsQuestion1_optionValue0=What is your e-mail password?
   ```
6. Save and close the file.

   In the example, when users access the Set Up Q & A Authentication page, Question 1 will not display a list of questions, but instead will display the question you specified in step 5.

Even though users must provide an answer to a particular question during Q & A setup, this does not mean that they must answer the question as part of a Q & A authentication.

To configure a question so that all users must answer it during authentication:

1. Open the **WXAuthQA** properties file.
2. Under QUESTION FIELD ATTRIBUTES, find the set of attributes that uses the same number as the attribute you chose in step 2 of the previous procedure.
3. Add the following attribute and value:

   ```
   tnQuestion#_required=true
   ```

   where # is the number of the attribute you that edited in the previous procedure.

   Adding the **_required** attribute and setting it to **true** forces the user to answer the question and verifies that the answer is correct.
4. Save and close the file.

Using the API to Verify Answers

Web Express provides an API that allows answers to be verified against data stored in a third-party database or an LDAP directory. For example, you might want a user to answer a question by providing data that is stored in a Human Resources database. Rather than duplicate this data in the RSA ACE/Server database, the Web Express API allows Web Express to pass the question and answer to the API call, which checks the Human Resources database and returns a value indicating a correct or an incorrect answer.

Regardless of the number of questions that use the API to verify answers, the API returns a single value that indicates that all answers are correct or that no answers are correct. Therefore, when you use the API to determine whether or not answers are correct, the code you write must determine how many correct answers constitute a good return value. When a good value is returned, all answers to questions that use the API are considered to be correct, and are counted against the question threshold value as correct. For example, consider the following scenario:

- Users must answer seven out of ten questions correctly.
- The answers to four of the questions are verified using the API, and your code specifies that the user must answer two of these questions correctly.

If the user answers two or more of these questions correctly, the API returns a good value, indicating that four questions have been answered correctly. Of the remaining six questions, the user must answer three correctly because the return value of the API counts as four correct answers against the question threshold of seven.

To use the API to verify all answers:

1. Open the **WXAuthQARegistration** properties file.
2. Set the values of the attributes that specify the questions to the text of the questions. You can use the default list method or the static question method of specifying questions.

   For more information, see the procedures in the preceding section, "Configuring the List of Questions."
3. Save and close the file.
4. Copy the **CustomAPI.java** file from the **\supported** directory on the RSA SecurID Web Express 1.2 CD to your PC, or if you have already copied and edited the file, use the edited copy.
5. Open the file and modify the **APIValiduateQuestionsAndAnswers** method.

   For information on modifying the API, see the appendix "Using the RSA SecurID Web Express APIs" in this book, and the comments in the **CustomAPI.java** file.
6. Compile and test your changes in a test environment.
7. When you finish testing, copy the modified **CustomAPI.class** file to the **RSAWebExpress\JRun\servers\default\rsaswe\WEB-INF\classes** directory.

8. Restart the JRun server.

   The API returns APPROVE_QA when the user answers enough questions correctly or REJECT_QA when the user does not answer enough questions correctly.

To use the API to verify some of the answers:

1. Open the **WXAuthQARegistration** properties file.
2. Decide how many of the questions will be verified by the API.
3. Set the values of the attributes that specify the questions to the text of the questions. You can use the default list method or the static question method of specifying questions.

   For more information, see the procedures in the preceding section, "Configuring the List of Questions."
4. Save and close the file.
5. Open the **WXAuthQA** properties file.
6. Add the following **_api** attribute to the set of attributes that specify any question that is verified by the API:

   ```
   tnQuestion#_api=YES
   ```

   where # matches the number used by the other attributes in the set.
7. Save and close the file.
8. Copy the **CustomAPI.java** file from the **\supported** directory on the RSA SecurID Web Express 1.2 CD to your PC, or if you have already copied and edited the file, use the edited copy.
9. Open the file and modify the **APIValiduateQuestionsAndAnswers** method.

   For information on modifying the API, see the appendix "Using the RSA SecurID Web Express APIs" in this book, and the comments in the **CustomAPI.java** file.
10. Compile and test your changes in a test environment.
11. When you finish testing, copy the modified **CustomAPI.class** file to the **RSAWebExpress\JRun\servers\default\rsaswe\WEB-INF\classes** directory.

    The API returns APPROVE_QA_WITH_WX_TEST when the user answers enough questions correctly and REJECT_QA when the user does not answer enough questions correctly. In the former case, Web Express then checks the remaining answers against the RSA ACE/Server database.

Storing Questions and Answers

By default, Web Express stores the questions and answers in user extension fields in the requester's user record in the RSA ACE/Server database. If Web Express uses the API to verify an answer, only the question is stored in a token extension field; the answer is stored in the third-party data source. In this case, answers provided by the user during Q & A setup are not stored by Web Express.

You can configure Web Express to store the answers in clear text or in encrypted format in the RSA ACE/Server database. By default, the answers are encrypted.

To configure the storage method:

1. On the Admin Home page, click **Configure System** > **System Setup** > **Workflow & User Account**.
2. Under Question and Answer Authentication
   - To store the answers in user-readable format, check the **Stored Answer Format** checkbox.
   - To store the answers in encrypted format, clear the **Stored Answer Format** checkbox.
3. At the bottom of the page, click **Save**.

Specifying Authentication Methods on a Page

You can configure Web Express to require requesters to authenticate before accessing certain pages. For example, the Change PIN page is protected by RSA SecurID authentication or Q&A authentication by default. The authentication methods used to protect pages are specified by the value of the **authMethods** attribute in the GENERAL ATTRIBUTES section of the page's properties file.

Note: If a page does not have an **authMethods** attribute, you can add one to the GENERAL ATTRIBUTES section of the page's properties file.

There are three possible values for this attribute:

- **SecurID**, which requires the user to authenticate using an RSA SecurID token or user password.
- **QA**, which requires the user to answer a set of questions.
- **Password**, which requires a Web Express administrator to enter the administrator password. This method protects access to the Admin Home page. Only Web Express administrators can access pages protected by this method.

  The administrator password is specified during installation of Web Express or through the **Edit Administrators** page. For more information, see the Web Express Help.

You can protect a page with any of these methods. To protect a page with multiple methods, and thereby allow the requester to choose between them, separate the values with a semi-colon. For example, to protect a page with SecurID and Q & A authentication, set the attribute to the following:

```
authMethods=SecurID;QA
```

The Activate Token Page

Once requesters receive the activation code, they must access the Activate Token page and provide their user IDs, activation codes, activation passwords, if configured, and additional information that you require in any custom fields. If you allow users to self-assign tokens, possible only with hardware tokens, they must also enter the serial number of their token in the Token Serial Number field.

Requiring the Activation Password

This field displays when you have configured the Request Token page to require requesters to enter an Activation Password. For more information, see "Enabling and Disabling the Activation Password" on page 34 and the Help topic, "Customize Web Express for Users."

Allowing Users to Self-Assign Tokens

When requesters specify a serial number in the Token Serial Number field, Web Express uses the input to assign the token to the requester. This field is editable only when you allow requesters to self-assign tokens during activation. When you configure Web Express so that the Distributor manually assigns tokens to users, the Activate Token page displays the serial number of the assigned token in the **Token Serial Number** field.

To allow users to self-assign tokens:

1. On the Admin Home page, click **Configure System** > **System Setup** > **Workflow & User Account**.
2. In the **Hardware Distribution** drop-down list, select either **No Distributor. Requesters get unassigned tokens some other way** or **Distributor deploys unassigned tokens**.
3. At the bottom of the page, click **Save**.

The Replace Token Page

The Replace Token page allows users with expiring tokens to activate new, replacement tokens under the following conditions:

- The user's current token is enabled and expiring within the number of days configured as the **Token Replacement Cutoff** in the Workflow & User Account page in Web Express.
- The user has been given an unassigned token to be used as the replacement token.

To configure the Token Replacement Cutoff:

1. In the Web Express Admin Home page, click **Configure System** > **System Setup** > **Workflow & User Account**.
2. Under User Account Management, enter the number of days left before expiration in the **Token Replacement Cutoff** field.
3. At the bottom of the page, click **Save**.

5 Troubleshooting

This chapter offers solutions to common problems you may experience with Web Express.

Problems Starting RSA SecurID Web Express

RSA SecurID Web Express Fails to Start

If Web Express fails to start:

- Make sure the required information was entered during installation. For a list of required information, see "Step 5: Collect Required Information" on page 12. To determine whether the information was entered, check the **config.properties** file in the ***Web_Express_installation_directory*\RSAWebExpress\JRun \servers\default\rsawe\WEB-INF\config** directory.

  If the required information was not entered during installation, uninstall Web Express, and reinstall it, providing the information when you are prompted for it. For instructions, see the appendix "Uninstalling RSA SecurID Web Express" in this book.

- Confirm that these settings are correct:

  – **ACE_SERVER.** The fully qualified host name of the RSA ACE/Server with which Web Express communicates.

  – **ACE_PORT.** The port number assigned to the ACE Comm Service. This port number must match the port number configured on the RSA ACE/Server.

  – **ACE_USERID.** The User ID of the Web Express administrator account on the RSA ACE/Server.

  – **ACE_PASSCODE.** The password for the Web Express administrator account on the RSA ACE/Server. This password must be static and have administrative privileges.

  These settings are specified during Web Express installation. However, they may have been changed on the RSA ACE/Server before they were updated in Web Express. For example, if the password assigned to the Web Express account on the RSA ACE/Server was changed before it was updated in Web Express, you will not be able to start Web Express.

To verify and edit the RSA ACE/Server settings:

1. Open the **config.properties** file in your default text editor.

   The file is in the ***Web_Express_installation_directory*\RSAWebExpress \JRun\servers\default\rsawe\WEB-INF\config** directory.

2. Check the RSA ACE/Server values, and edit them if necessary.

   Follow the file conventions in the following section, "File Conventions."

   Note: To change the User ID and password, you must delete the SECURITY_BLOCK_DO_NOT_REMOVE line from the **config.properties** file. Deleting this line removes the encryption and shows the User ID and password in the clear. When you start Web Express, the User ID and password are re-encrypted.

3. Save the changes, and close the file.
4. Start Web Express.

File Conventions

Edit the RSA ACE/Server settings according to these conventions:

- Two-state settings have these characteristics:
  - Accept only **Yes** or **No** as values. These values are not case sensitive.
  - Default to **No** if no value is stated, or if a value other than **Yes** or **No** is stated.
- Values for settings other than two-state settings are case sensitive, and must be entered as they appear in this document.
- Spaces are allowed in value strings. It is not allowed in setting names or on either side of the equals sign (=).
- All instances of ***n*** must be replaced with an integer that is greater than or equal to one.
- All instances of ***nn*** must be replaced with a two-digit integer. If the value is less than 10, it must include a leading zero (for example, **08**).

Errors Occur While Starting RSA SecurID Web Express

These errors may occur when you attempt to start Web Express.

Error Message	Possible Cause	Solution
"Failed to create ACEObjectPool. Failed to connect to the RSA ACE/Server <*server name*> with port=5570. ACE Error: null"	The RSA ACE/Server Quick Admin daemon is not started on the RSA ACE/Server.	Start the RSA ACE/Server Quick Admin daemon on the RSA ACE/Server. For instructions, see the Help topic "Start the RSA ACE/Server Quick Admin Daemon."

Error Message	Possible Cause	Solution
"Failed to create ACEObjectPool. Failed to connect to the RSA ACE/Server <*server name*> with port=5570. ACE Error: Verify protocol version 75 not supported by the ACS/ACD"	The web server name and IP address are not in the **hosts.conf** file. This file is in the ***RSA ACE/Server_ installation_directory*****\prog** directory on your RSA ACE/Server.	Add the web server name and the IP address to the **hosts.conf** file. For details, see "Step 1: Configure RSA ACE/Server to Work With RSA SecurID Web Express" on page 10.
"Failed to create ACEObjectPool. Unable to send passcode to the RSA ACE/Server. Access is denied"	• The ACE Communication client is not open to all locally known users, or the RSA ACE/Server administrator does not have access to the specific client on the RSA ACE/Server. • The ACE Password is not set properly in either the **config.properties** file or on the RSA ACE/Server.	• In the RSA ACE/Server, enter the Web Express user record as a known user for the ACE client, or select "Make open to all locally known users" in the RSA ACE/Server. You created the Web Express user record in "Step 2: Prepare the RSA ACE/Server Database to Communicate With RSA SecurID Web Express" on page 11. For information about editing the account on the RSA ACE/Server, see the RSA ACE/Server documentation. • Confirm RSA ACE/Server settings in the **config.properties** file. For instructions, see "RSA SecurID Web Express Fails to Start" on page 49.
"Node verification failure" This error appears in the RSA ACE/Server activity log.	The node secret on the RSA ACE/Server does not match the node secret on the Web Express server.	Delete the node secret on the RSA ACE/Server and on Web Express. On the RSA ACE/Server, the file is in the **/certs** directory. On Web Express, the node secret is the ***Web_Express_installation _directory*****\RSAWebExpress\JRun \servers\default\rsawe\WEB-INF \cert** directory.

Problems Connecting to the Web Express Administrator Pages

If you cannot connect to the Web Express administrator pages, determine whether JRun is running.

To test JRun:

1. Go to **http://*hostname*/demo** where ***hostname*** is the name of the Web Express host.
2. When you see the message **Servlet Ran Successfully!**, close the Sample Servlets window.

   If the JRun Demo does not run successfully, check the log files in the ***Web_Express_installation_directory*\RSAWebExpress \JRun\logs** directory for errors.

Problems Communicating With the RSA ACE/Server

The RSA ACE/Server Quick Admin daemon enables communication between Web Express and the RSA ACE/Server. If the RSA ACE/Server Quick Admin daemon fails to start, verify that

- The RSA ACE/Server Quick Admin daemon is running. Click **Start** > **Settings** > **Control Panel** > **Administrative Tools** > **Services**. The status of **ACE Comm Service** should be **Started**. If it is not started, click **Web Admin Daemon**, and click **Start**.
- The **hosts.conf** file in the RSA ACE/Server **ace\prog** directory contains the fully qualified name and IP address of the web server. For details, see "Step 1: Configure RSA ACE/Server to Work With RSA SecurID Web Express" on page 10.
- The RSA ACE/Server and the Web Express software are running the RSA ACE/Server Quick Admin daemon on the same port. The default port number is **5570**.
- The RSA ACE/Server username assigned to Web Express has been added to the RSA ACE/Server. The RSA ACE/Server username is the ACE_USERID parameter in the Web Express **config.properties** file. For instructions on adding the RSA ACE/Server username to the RSA ACE/Server, see the RSA ACE/Server documentation.
- The RSA ACE/Server username assigned to Web Express has administrator privileges. For information, see the RSA ACE/Server documentation.
- The RSA ACE/Server username assigned to Web Express has been assigned an eight-digit, numeric, static password, and this password is the same on the web server. For information, see "RSA SecurID Web Express Fails to Start" on page 49.
- The RSA ACE/Server settings in the **config.properties** file are correct. For details, see "RSA SecurID Web Express Fails to Start" on page 49.

- The RSA ACE/Server name is added to the client list, and all new users are allowed access to it. For details, see the RSA ACE/Server documentation.
- The web server is added to the client list, and all new users are allowed access to it. For details, see the RSA ACE/Server documentation.
- The **sdconf.rec** file for the RSA ACE/Server is in the ***Web_Express_installation_directory*\RSAWebExpress\JRun\servers\default \rsawe\WEB-INF\cert** directory on the Web Express host.
- The certificate files **sdti.cer** and **server.cer** on the Web Express host match the certificate files on the RSA ACE/Server.

Problems Processing Approver and Distributor Requests

Problem	Possible Cause	Solution
An Approver approves a request, but it is marked as rejected and moved to the Archive database.	• An error occurred while creating or updating the user record on the RSA ACE/Server. • The Access Request group the requester selected does not match any of the groups in the RSA ACE/Server database. • A user record with the same username was already created on the RSA ACE/Server. This applies only if Web Express is configured to reject requests from users who already have records in the RSA ACE/Server database.	• Confirm that the connection to the RSA ACE/Server is correct. For information, see the Help topic "Edit Connections." • Confirm that the Web Express user record has Administrator privileges on the RSA ACE/Server. You created the Web Express user record in "Step 2: Prepare the RSA ACE/Server Database to Communicate With RSA SecurID Web Express" on page 11. • Confirm Access Request List Settings in the Token Request page properties file. For details, see the chapter "Customizing RSA SecurID Web Express" in this book. • Check the record for the rejected request in the Archive database. If you see the message, "Unable to add new user, *username* already exists in ACE/Server," have the user submit a new request with a different username.

Problems Opening the JRun Management Console

If the JRun Management Console does not open after you install Web Express, perform these steps:

1. Click **Start > Programs > RSA SecurID Web Express > JRun Admin Server**.
2. Click **Start** > **Programs** > **RSA SecurID Web Express**> **JRun Management Console**.

If the JRun Management Console still does not open, see the *JRun Setup Guide*. You can get this guide from the Macromedia web site.

Problems Sending e-Mail

If an attempt to send Web Express e-mail results in a parsing error, make sure the e-mail From field is configured. You must configure the From field in Web Express e-mails. Otherwise, some mail systems may be unable to process the mail.

A Using the RSA SecurID Web Express APIs

RSA SecurID Web Express ships with a set of customizable APIs. This appendix

- Describes the APIs and gives examples of how you can use the APIs within your business process
- Gives instructions for implementing the APIs

For technical details and sample source code, see the **CustomAPI.java** file in the **\support** directory on the RSA SecurID Web Express 1.2 CD.

Using the APIs Within Your Business Process

This section describes the APIs and gives examples of how you can use them in your business process.

Validate Request

The Validate Request API (**APIValidateRequest**) verifies token request data by matching it against data in a customer-defined data source. When the Validate Request API is called, the values of the token request are passed to the API. These values can be used to programmatically validate the request.

This API enables you to

- Automate the approval process.
- Pre-validate a request before the Approver receives it.

  For an example, see "External Data Distribution API" on page 56.
- Enable the Approver to compare incorrect information that the user enters with correct information from a data source, and make appropriate changes.

  For example, you can look up a user's cost center (using the User ID as a search key), and reject the request if the user has specified the wrong cost center.

Return Values

The Validate Request API returns these values:

API_APPROVE_WITH_MANAGER. The API finds no reason to reject the request. The API also returns the validation data for the request fields. The final decision to approve or reject the request is left to the Approver.

API_APPROVE_WITHOUT_MANAGER. The API approves the request. There is no action left for the Approver.

API_REJECT. The API rejects the request. There is no action left for the Approver.

Data Import API

The Data Import API (**APIUpdateRequestFields**) retrieves data from an external data source and imports it into Web Express. In addition, upon approval of a token request, requester data is written into the RSA ACE/Server database.

When the Data Import API is called, the values of the token request are passed to the API. The fields in the token request can then be used to retrieve additional information about the user from an external corporate data source. This data is then passed back to Web Express, where it is added to the original request data.

For example, suppose the token request contains a custom field for employee ID. The API can use the employee ID to look up data in an external corporate data source. Additional fields in the token request (such as First Name, Last Name, E-mail) can then be filled in with the data from the external data source. The request process continues with the new data in the request fields.

Return Values

UPDATE_FIELDS. The API tells Web Express to update the request fields.

Important: The update function updates *all* the fields in the request. An empty string results in an empty field. If you want to change only certain fields, you must put the data back into the fields you are not updating. For example, if the Cost Center field is left empty, Web Express puts the request into a nonexistent “blank” Cost Center and the Approver never knows that the request was made.

DO_NOT_UPDATE_FIELDS. The API tells Web Express not to update the request fields.

External Data Distribution API

The External Data Distribution API (**APIValidateReqAfterMgrAprl**) distributes information about approved requesters to external data sources. This API enables you to

- Update a corporate data source after a request has been accepted.
- Accept or reject the request after a secondary validation process.

Before a request is forwarded for approval, the Validate Request and Data Import APIs are called. Once a request has been approved, the External Data Distribution API is called.

The External Data Distribution API works similarly to the Validate Request API. However, the External Data Distribution API is called *after* a request has been accepted by an Approver. The data from the approved request can be used to update an external corporate data source. In addition, this API enables you to reject a request that has been initially approved.

For example, suppose the token request contains a custom field for corporate ID. The API can update an external corporate data source to indicate that the user who is represented by that corporate ID now has an RSA SecurID token.

Return Values

APPROVE_REQUEST. The API returns this value if the request is awaiting deployment. The API finds no reason to reject the request, but the final decision to approve or reject the request is left to the Approver.

REJECT_REQUEST. The API returns this value if the request is rejected. There is no action left for the Approver.

Validate Activation API

The Validate Activation API (**APIValidateActivation**) validates information requesters submit to activate their tokens by matching it against data in a customer-defined data source. When the Validate Activation API is called, the values submitted to activate a token are passed to the API.

This API enables you to ensure that user data (such as User IDs or passwords) maintained in more than one location match. For example, suppose your company uses LDAP passwords that are managed in an LDAP database, and you want employees to use the same password for all applications. You can configure the Validate Activation API to check the password the requester submits against the password in the LDAP database.

Return Values

ACTIVATION_VALID. The API finds the data in the customized fields valid and activates the token.

ACTIVATION_INVALID. The API finds the data in the customized fields invalid and rejects the activation request.

Get PIN Parameters

The Get PIN Parameters API (**APIGetPinParams**) collects a set of PIN requirements (for example, the minimum and maximum number of characters, whether PINs can be alphanumeric, whether PINs must be system-generated or created by users). This API is called by the Change PIN API, described in the next section.

Return Values

APPROVE_CHANGE_PIN. The API tells Web Express there are PIN requirements provided in a hash map, and to apply the requirements when guiding the user through the change PIN process.

REJECT_CHANGE_PIN. The API tells the Change PIN API to reject the request to change the PIN.

Change PIN API

The Change PIN API (**APIChangePIN**) enables users to use Web Express to change the PINs they use to access other applications (for example, LDAP databases, voicemail, Windows). This API calls the Get PIN Parameters API, described in the preceding section.

Return Values

APPROVE_CHANGE_PIN. The API tells Web Express the PIN was successfully changed.

REJECT_CHANGE_PIN. The API tells Web Express to reject the request to change the PIN.

Validate Question and Answer API

The Validate Question and Answer API (**APIValidateQ&A**) verifies Question and Answer authentication information requesters submit by matching it against data in a customer-defined data source.

For example, suppose you configure Web Express to ask requesters for their LDAP passwords as part of Question and Answer authentication. When a requester submits Question and Answer data, the password data is passed to the Validate Question and Answer API. The API compares the password the requester submitted to the password stored in the corporate LDAP database.

Return Values

APPROVE_QA. The API tells Web Express to approve the request without testing.

APPROVE_QA_WITH_WX_TEST. The API tells Web Express to test the request before approving it.

REJECT_QA. The API tells Web Express to reject the request.

Implementing the APIs

This section explains how to implement the APIs.

Note: If you enable an API after you have begun using Web Express, requests already in process may not be recognized by the API. The point at which the API is called determines whether the requests are recognized. For example, if you enable an API that is called during approval, requests that were approved before the API was enabled are not affected.

To implement one or more of the APIs:

1. Copy the **CustomAPI.java** file from the **\support** directory on the RSA SecurID Web Express 1.2 CD to a directory on your development machine.

2. In Windows Explorer, right-click the name of the file, click **Properties**, and clear the read-only attribute setting.
3. Modify the contents of the file to suit your environment.
4. Compile the modified **CustomAPI.java** file.
5. Create a new class file.
6. Replace the **CustomAPI.class** file in the ***Web_Express_installation_directory*\RSAWebExpress \JRun\servers\default\rsawe\WEB-INF\classes** directory with the class file you created in the previous step.
7. Use Web Express to enable the API.

   For instructions, see the Help topic "Enable RSA SecurID Web Express APIs."
8. Stop and restart Web Express.

B Upgrading RSA SecurID Web Express

This appendix explains how to upgrade from Web Express 1.0 and 1.1 to Web Express 1.2.

If you want to install Web Express for the first time, see the chapter "Installing and Setting Up RSA SecurID Web Express" in this book.

Upgrading Web Express

- Puts configuration properties files from the previous installation into the new installation
- Configures the existing databases to work with Web Express 1.2

Upgrading Web Express does not uninstall the former Web Express installation.

Important: Do not uninstall the former installation of Web Express ***before*** copying the **config.properties** file and backing up the database or ***after*** installing the upgrade version.

If you want to uninstall the former installation of Web Express, you must follow this order:

1. Copy the **config.properties** file, and back up the database. The procedure "To upgrade Web Express" on this page guides you through these tasks.
2. Uninstall the former version of Web Express.
3. Reconnect the database drivers to the databases.
4. Install the new version of Web Express.

You can perform these types of installations:

Full installation. Two instances of JRun exist on the Web Express host. The former Web Express installation remains available.

Important: If you perform the upgrade as a full installation, make sure the ports used for the former installation and the ports used for the upgrade installation do not conflict.

Partial installation. Web Express is installed into an existing installation of JRun. The former Web Express installation becomes unavailable.

To upgrade Web Express:

1. In the current version of Web Express, process all outstanding requests.
2. Stop the current version of Web Express.
3. Back up your current Web Express database.

4. If you protected the Web Express administrator pages with an application such as RSA WebID, unprotect the pages.
5. Create a temporary directory on the machine that will host the upgrade of Web Express.
6. Copy the **config.properties** file from the ***JRun_installation_directory*\servers\default\wbur-app\WEB-INF\config** directory in the previous installation to the directory you created in the preceding step.
7. Copy these files into the directory you created in step 4:
   - **sdti.cer**
   - **server.cer**
   - **sdconf.rec**

   These files are in the **ace\data** directory on your Primary RSA ACE/Server.
8. Insert the RSA SecurID Web Express 1.2 CD, and follow the instructions on the screens.

Important: If you are using an Access database, and uninstall the former version of Web Express, you need to reconfigure the database. For instructions see, "Step 3: Configure the Web Express Database" on page 11.

C Using a Third-Party Database

By default, RSA SecurID Web Express is configured to work with the Microsoft Access database for active requests and archived requests. However, you can use a database other than Microsoft Access.

This appendix explains how to configure a third-party database to work with Web Express.

Configure Your Database to Work With RSA SecurID Web Express

To configure your database to work with Web Express:

CAUTION: Only highly experienced database administrators should perform this procedure.

1. Make sure the database software is installed correctly and the database services are running.
2. Create two databases; one for transactions, and one for archive data.
3. Copy the JDBC database driver files to the ***Web_Express_installation_directory*\RSAWebExpress \JRun\servers\default\rsawe\WEB-INF\lib** directory.

   The driver files have either **.jar** or **.zip** extensions. Contact your database vendor to find out which files you need.
4. Use the tools provided with your database to execute each of these scripts against each of the databases you created in step 2:
   - **admins.sql**
   - **requests.sql**
   - **costcenter.sql**

   Note: These scripts are in the **supported\sql_scripts** directory on the RSA SecurID Web Express 1.2 CD. You may need to modify the scripts to work with your database. See the following section, "Modifying the Scripts."

   This creates these tables in each of the databases:
   - ADMINS
   - REQUESTS
   - COSTCENTER

Modifying the Scripts

Some servers may require these modifications:

- Remove the "COMMIT;" line at the end of each script before running it.
- If you use the script **requests.sql** to create the REQUESTS tables, change the data type and precision in the REQUEST_NUMBER line to:

  ```
  REQUEST_NUMBER NUMERIC(28,0)
  ```

- The **admins.sql** script creates the ADMIN table.
- The **cost_center.sql** script creates the cost center table, and adds sample cost center data to the cost center table.

Important: Before you can edit the scripts, for each file, right-click the name of the file in Windows Explorer, click **Properties**, and clear the read-only setting.

You must edit the sample cost centers to match your own environment, then create Approvers and Distributors and assign them to the new cost centers. For instructions, see the Help topics “Manage Cost Centers” and “Manage Administrators.”

D Uninstalling RSA SecurID Web Express

This appendix tells how to uninstall Web Express 1.2. If you want to uninstall a former version, see the documentation provided with that version of Web Express.

Important: If you have other Java applications running on the web server, do not remove JRun or JRE.

To uninstall Web Express:

1. Stop Web Express.
2. Click **Start** > **Settings** > **Control Panel** > **Add/Remove Programs**.
3. Remove **RSA SecurID Web Express**.

Index