Security Methods for the Acrobat Family of Products

A guide for administrators describing password security, certificate security, and Livecycle (Policy Server) security, including configuration and deployment details

Acrobat® Family of Products (All versions)
Modification date: 5/16/12

Document Security in the Adobe® Acrobat Family of Products.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA. Notice to U.S. Government End Users. The Software and Documentation are “Commercial Items,” as that term is defined at 48 C.F.R. §2.101, consisting of “Commercial Computer Software” and “Commercial Computer Software Documentation,” as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contents

1 Introduction 6

1.1 What's Changed with 10.0 6

1.1.1 Password Security 6

1.1.2 Rights Management and LiveCycle 6

1.2 Quick Key 8

1.3 Encryption Algorithms 8

1.4 Other resources 9

2 Document Security Basics 11

2.1 Security Method Basics 11

2.1.1 Choosing a Security Method Type 12

2.1.2 Security Policies 13

2.1.3 Security Methods and Encryption 13

2.1.3.1 Encryption Workflow 13

2.1.3.2 Choosing What to Encrypt 14

2.1.3.3 Choosing an Algorithm and Compatibility Level 14

2.1.4 Security Methods and Permissions 15

2.1.4.1 Permissions Workflow 15

2.1.4.2 Permissions Options 16

2.1.5 Associating an Action with a Security Method 17

2.2 Changing and Viewing Security Settings 17

2.2.1 Viewing Document Encryption and Permission Settings 18

2.2.2 Viewing Document Restrictions 19

2.2.3 Viewing Security Settings in a Browser 20

2.2.4 Changing the Security Method Type 20

2.2.5 Editing Security Method Settings 21

2.2.6 Removing Document Security 21

2.3 Security Policies: Reusable Security Settings 21

2.3.1 Creating Security Policies with Policy Manager 22

2.3.2 Applying a Security Policy to a Document 23

2.3.3 Viewing a Security Policy 24

2.3.4 Copying a Security Policy 24

2.3.5 Editing a Security Policy 24

2.3.6 Making a Security Policy Favorite 25

2.3.7 Refreshing the Security Policy List 25

2.3.8 Deleting a Security Policy 25

2.4 Envelopes 26

3 Password Security 28

3.1 Creating Password Security Settings 28

3.1.1 Creating a Reusable Password Security Policy 28

3.1.2 Creating Password Security for One-Time Use 31

3.2 Opening a Password-Protected Document 33

3.3 Removing Password Security 33

3.4 Password Recovery 34

4 Certificate Security 35
4.1 Setting up the Certificate Security Environment 35
4.1.1 Accessing the Windows Certificate Store 35
4.1.2 Selecting a Certificate to Use for Encryption 36
4.2 Working with Groups of Contacts 37
4.2.1 Creating a Group 37
4.2.2 Adding or Removing Group Contacts 38
4.3 Creating Certificate Security Settings 38
4.3.1 Creating a Reusable Certificate Security Policy 38
4.3.2 Creating Certificate Security for the Current Document 42
4.3.3 Applying a Certificate Security Policy 46
4.3.4 Applying a Certificate Security to a Group 46
4.3.5 Opening a Certificate-Protected Document 47
5 LiveCycle Rights Management Server Security 48
5.1 Configuring Servers 48
5.1.1 Importing ALCRMS Settings 49
5.1.1.1 Security Settings Import 49
5.1.1.2 FDF File Import 49
5.1.2 Configuring ALCRMS Settings Manually 49
5.1.3 Managing your ALCRMS Account 50
5.2 Working with Documents and ALCRMS Policies 50
5.2.1 Creating an ALCRMS Security Policy 50
5.2.2 Applying ALCRMS Security 51
5.2.3 Synchronizing a Document for Offline Use 52
5.2.4 Revoking a Document 52
5.2.5 View a Document’s Audit History 52
6 Registry and plist Settings 53
6.1 Setting Basic Client and Workflow Preferences 53
6.1.1 Preventing End-User Modification 53
6.1.2 Importing Security Settings 56
6.1.3 Digital ID Management 58
6.1.3.1 Specifying extended certificate information 58
6.1.3.2 Allowing Self Sign Digital IDs 58
6.1.3.3 PKCS#11 Configuration 59
6.1.3.4 Roaming ID Configuration 59
6.1.3.5 Setting Digital ID Defaults 63
6.1.3.6 Digital ID Import and Export Paths 64
6.1.4 Certificate Management 65
6.1.4.1 Windows Integration 65
6.1.4.2 Trusted Identity List Configuration 66
6.1.4.3 Custom Certificate Preferences 67
6.1.5 Custom Security Handler Preferences 71
6.1.6 ASPKI Service Provider Interfaces 73
6.1.6.1 Revocation Checker Providers 73
6.1.6.2 Signature Validation Directory Providers 74
6.1.6.3 Roaming ID Authentication Mechanisms 75
6.1.7 FDF Import and Export 76
6.1.8 Security Settings Console 77

6.1.9 Displaying All Chains in the Certificate Viewer 77
6.1.10 Configuring Password Caching 78
6.1.11 Turning on FIPS Mode 78
6.2 Content Security 79
6.2.1 Configuring Certificate Handling for Certificate Security 80
6.2.2 Requiring FIPS-Compliant Encryption Algorithms 80
6.2.3 Setting up Security Policy Favorites 80
6.2.4 Configuring LiveCycle Right Management Servers 81
6.2.5 Security Envelopes 82
7 Supported Standards and RFCs 83
8 Glossary of Security Feature Terms 85
9 Index 93

1 Introduction

This guide describes the technical details of the Acrobat family of products' document security-related features.

Audience

While this document does contain some end user "how to use the feature" content, it primarily provides details not found in end user help or in the SDK. The primary focus here is to help administrators and other business users set up and maintain secure PDF workflows. Potential audiences might include:

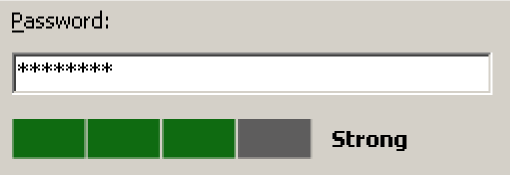
- Administrators who configure and deploy applications across their enterprise.
- Developers who need registry level detail to augment SDK information about creating custom plug-ins and handlers that use Acrobat's security features.
- Other business users that need advanced knowledge of signature workflows.

1.1 What's Changed with 10.0

1.1.1 Password Security

- The option to set backward compatibility with Acrobat 9.x has been removed. Compatibility with other version is still possible. This does not prevent the use of 9.x products with password protected files, but it does assure that the strongest encryption is used when possible.
- The method for encrypting the stored password has been strengthened to reduce the efficiency multi threaded attacks.
- A password strength meter has been added to the UI to help the user select and use stronger passwords.

Figure 1 Password strength meter



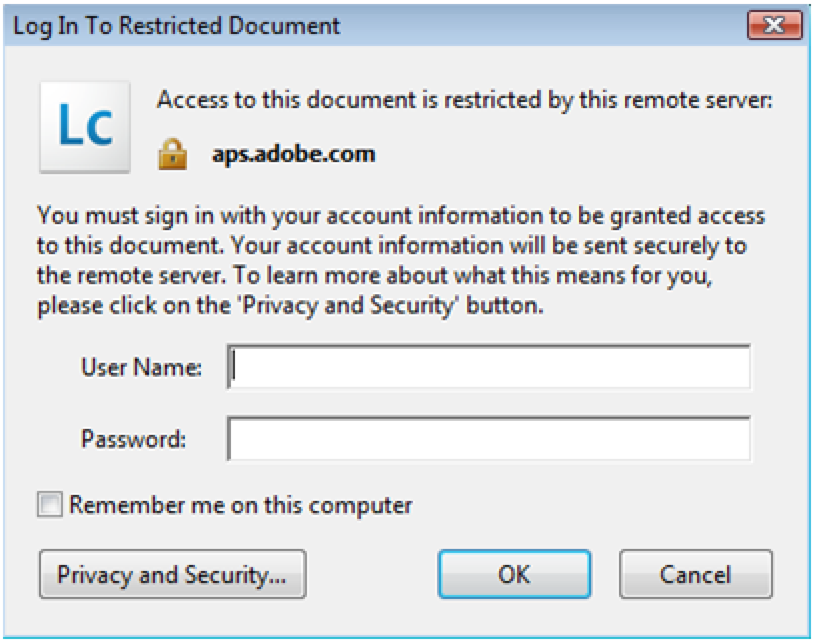
Password: *********

Strong

1.1.2 Rights Management and LiveCycle

When logging in to a LiveCycle server to access a restricted document, users have the option to save their login information locally by checking the **Remember me on this computer** (Figure 2). To improve security of the server as well as the application and protect end-users' personal data, the UI has been changed to allow the user to clear the stored login credential. To clear this data, choose **Edit > Preferences > Security > Clear remembered account information** (Figure 3).

Figure 2 LiveCycle log in



Log In To Restricted Document

Access to this document is restricted by this remote server:

aps.adobe.com

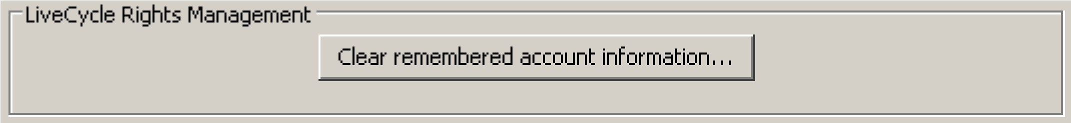
You must sign in with your account information to be granted access to this document. Your account information will be sent securely to the remote server. To learn more about what this means for you, please click on the 'Privacy and Security' button.

User Name:

Password:

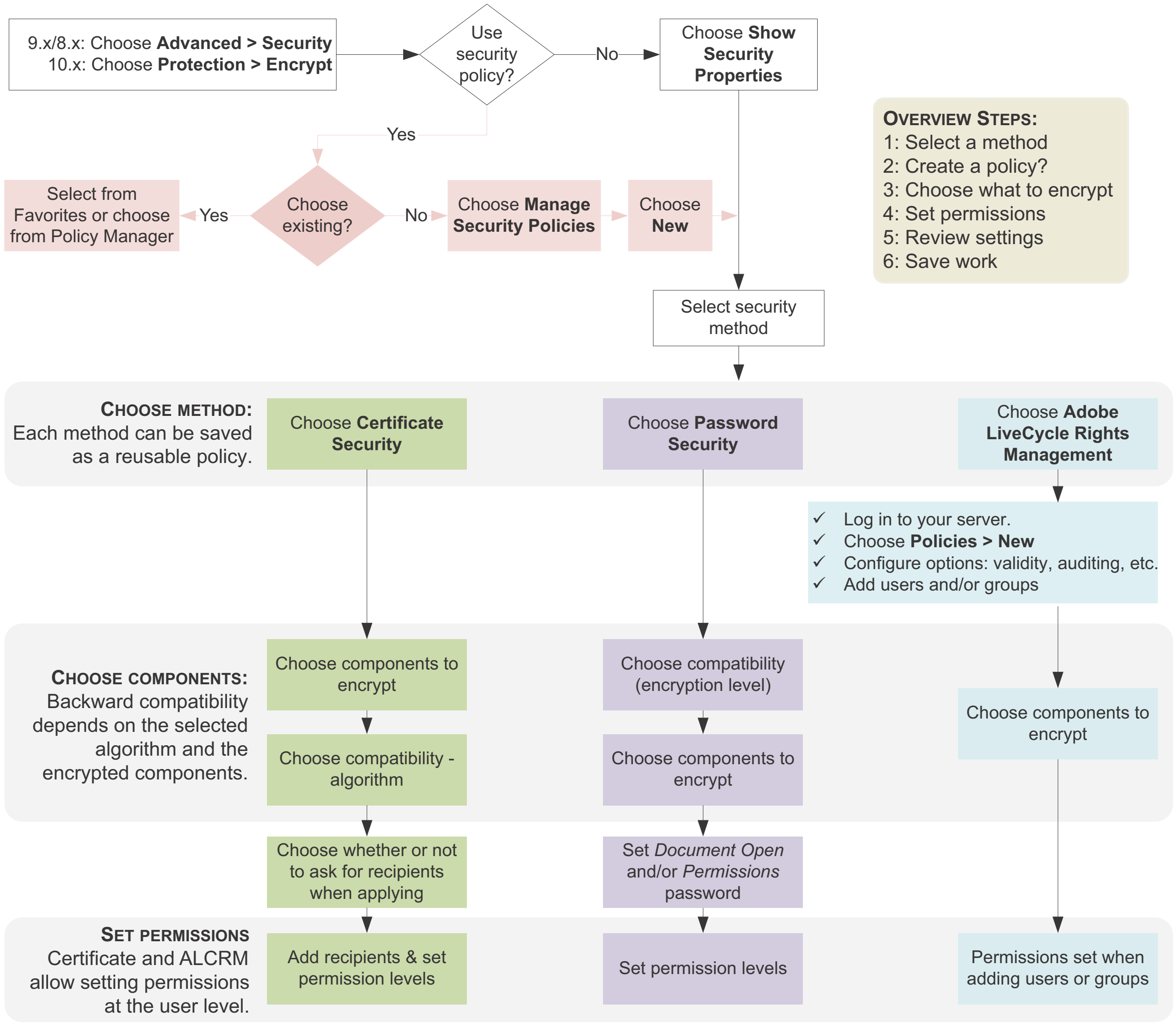
□ Remember me on this computer

Figure 3 LiveCycle: Clear login credential



LiveCycle Rights Management
Clear remembered account information...

1.2 Quick Key


9.x/8.x: Choose **Advanced > Security**
10.x: Choose **Protection > Encrypt**
Use security policy?
No
Choose **Show Security Properties**
Yes
Select from Favorites or choose from Policy Manager
Yes
Choose existing?
No
Choose **Manage Security Policies**
Choose **New**
Select security method
Choose method: Each method can be saved as a reusable policy.
Choose **Certificate Security**
Choose **Password Security**
Choose **Adobe LiveCycle Rights Management**
✓ Log in to your server.
✓ Choose **Policies > New**
✓ Configure options: validity, auditing, etc.
✓ Add users and/or groups
Choose components: Backward compatibility depends on the selected algorithm and the encrypted components.
Choose components to encrypt
Choose compatibility - algorithm
Choose compatibility (encryption level)
Choose components to encrypt
Choose components to encrypt
Choose whether or not to ask for recipients when applying
Set *Document Open* and/or *Permissions* password
Set permissions Certificate and ALCRM allow setting permissions at the user level.
Add recipients & set permission levels
Set permission levels
Permissions set when adding users or groups

Overview Steps:
1: Select a method
2: Create a policy?
3: Choose what to encrypt
4: Set permissions
5: Review settings
6: Save work

Pros:
- ✓ No password to remember.
- ✓ Key not susceptible to brute force discovery and resides only on the recipients machine.
- ✓ Can encrypt documents for specific people.
- ✓ Can use certificates from trusted 3rd party.
- ✓ Specifies different permissions for users.
- ✓ Leverages LDAP for recipients & groups.

Cons:
- ✓ Users must have a digital ID.
- ✓ Requires distributing/managing digital IDs.
- ✓ Full support appears first in 6.0.

Pros:
- ✓ Backward-compatible to Acrobat 3.0 for certain encryption levels.
- ✓ Simple and easily understood.
- ✓ Share documents by sharing the password.
- ✓ Different open & permission password.

Cons:
- ✓ Password strength is critical.
- ✓ Users share the same permissions.
- ✓ Disabled when in FIPS mode.

Pros:
- ✓ Centralized policy administration.
- ✓ Document auditing.
- ✓ Allows setting permissions for separate tasks such as opening, editing, and so on.
- ✓ Can specify different permissions for users.
- ✓ Leverages LDAP for recipients & groups.
- ✓ Offline control: Can specify a validity time limit after which document expires and is locke

Cons:
- ✓ Requires a network connection, an administrator, and a LiveCycle Server.

1.3 Encryption Algorithms

Refer to the Document Security Encryption Quick Key at www.adobe.com/go/learn_acr_security_en.

1.4 Other resources

This guide provides technical details that are probably not of interest to the casual user. It is also not a developer document, and developers should refer to the SDK and its associated references and APIs. As you peruse this document, keep in mind that there are numerous resources out there, including forums and even video tutorials. Adobe is aggressively revamping many of its learning resources as Web 2.0 matures, and it may be that one of these sites would prove equally, if not more, useful:

- Documentation Library: www.adobe.com/go/learn_acr_security_en
- Developer documentation: http://www.adobe.com/go/acrobat_security.
- Admin and end user documents:
  - http://www.adobe.com/support/acrobat: A fabulous resource that is rapidly evolving into the primary location for tutorials, guides, videos, blogs, and other help.
  - http://www.adobe.com/support/livecycle
  - http://www.adobe.com/support/reader
- White papers/data sheets: http://www.adobe.com/security
- www.acrobatusers.com

Note: Table 1 shows a partial list of the documentation residing at the above locations.

Figure 4 Resource roadmap

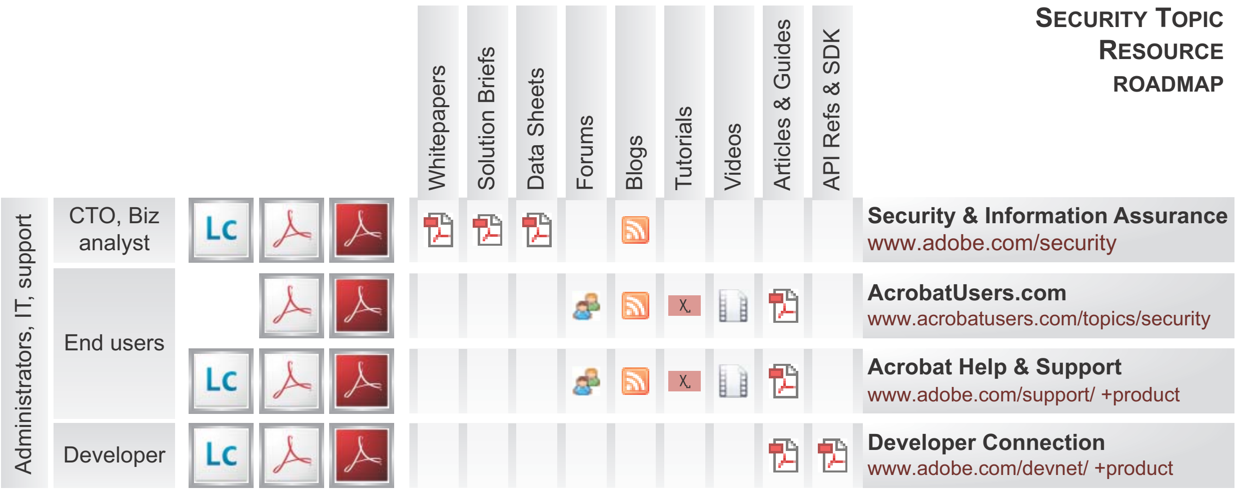


Table 1 Documentation related to security

Document	Audience	For information about
Acrobat SDK Documentation Roadmap	Developers	A guide to the documentation in the Adobe Acrobat SDK.
Acrobat and PDF Library API Reference	Developers	A description of the APIs for Acrobat and Adobe Reader® plug-ins, as well as for PDF Library applications.
JavaScript for Acrobat API Reference	Developers	A listing of the Acrobat JavaScript APIs.

Table 1 Documentation related to security

Document	Audience	For information about
Developing Acrobat Applications with JavaScript	Developers	Additional detail about the Acrobat JavaScript APIs.
PDF Reference 1.x	Developers	A detailed description of the PDF language.
FDF Data Exchange Specification	Developers	A object-level FDF file description. The files can be generated programmatically and used to share security-related data.
PDF Signature Build Dictionary Specification	Developers	Build properties for the PDF Reference's signature dictionary which provides interoperability details for 3rd party handlers.
Digital Signature Appearances	Developers & administrators	Guidelines for creating signatures programmatically.
Guidelines for Developing CSPs for Acrobat on Windows	Developers & administrators	Guidelines for developing a Cryptographic Service Provider for use with Acrobat® on the Windows® platform.
Enhanced Security in Adobe Acrobat 9 and Adobe Reader 9	Administrators & end users	X-domain configuration specifically and other aspects of the enhanced security feature generally.
Digital Signatures in the PDF Language	Anyone needing an overview	A generic description of how signature work in PDF.
Digital Signatures in Acrobat	Anyone needing an overview	A description of how signatures are implemented in Acrobat.

2 Document Security Basics

Security methods provide a mechanism for users to specify document encryption and permission settings. You can encrypt all or part of a document and limit user actions such as only allowing form field fill-in or preventing printing. Each security method offers a different set of benefits, so familiarize yourself with the pros and cons of each type before proceeding (Table 2). Additionally, you can reuse your security settings by saving them as a policy.

To learn about security methods, see the following:

- "Security Method Basics" on page 11: You should understand the options available for specifying the security type, the encryption and permissions options, and whether or not your security settings should be saved as a policy.
- "Changing and Viewing Security Settings" on page 17: Security settings can be changed at any time by the document author.
- "Security Policies: Reusable Security Settings" on page 21: If you would like to reuse your settings, save them as a policy.

For details about security method types, see the following:

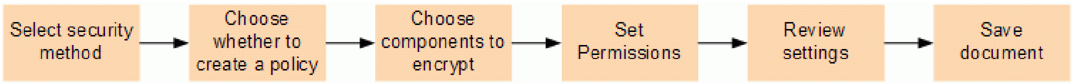
- Chapter 3, "Password Security": Use password security if document recipients do not have digital IDs or it is too cumbersome to collect their certificates.

  Tip: Password security is unavailable if your application is operating in FIPS mode. Trying to save a document with password security applied results in an alert stating that this security method uses a non-FIPS compliant algorithm.
- Chapter 4, "Certificate Security": Use certificate security if you can share digital ID certificates with workflow participants, need to configure different permissions for different users, or don't want to rely on shared passwords.
- Chapter 5, "LiveCycle Rights Management Server Security": If your company uses an Adobe LiveCycle Rights Management Server, use it to control document access and view audit trails.

2.1 Security Method Basics

Security is often added to documents to limit viewing, editing, printing, and other features to only those users that have the required password, a digital ID, or access to an Adobe LiveCycle Rights Management Server. Acrobat's default security methods not only protect document content from unauthorized access, but also allow users to specify encryption levels and permission settings. At a high level, adding security to a document involves selecting a security type, configuring encryption and permissions, and then saving the document (Figure 5).

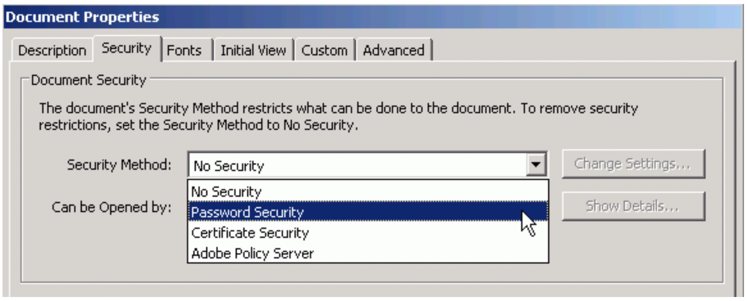
Figure 5 Security method workflow



Unless saved as a policy, security settings are document-specific and can not be applied to other documents. Security can be added to a document through two main methods:

- Create new settings that may or may not be saved as a policy: Choose **File > Properties > Security tab**, and then select and configure a method (Figure 6). Both certificate and ALCRMS security allows the user to save the settings as a policy.

Figure 6 Security method selection



The document's Security Method restricts what can be done to the document. To remove security restrictions, set the Security Method to No Security.

Security Method: No Security

- Create a new security policy with the Policy Manager: Choose **Tools** > **Protection** > **Encrypt** > **Manage**. When the Policy Manager opens, choose **New**, select a policy type, and configure it (Figure 16).

2.1.1 Choosing a Security Method Type

See also the Quick Key.

While custom security handlers may be installed, the default security methods provide a wide range of robust options. There are a number of reasons to choose one security type over another, but all methods let the user specify encryption algorithms, what document components to encrypt, and what permissions should be granted to whom. Selecting a security type can involve an analysis of each method's pros and cons (Table 2) or each security method's basic features:

- **Password security**: Password security provides a simple way to share documents among users where sharing passwords is possible or when high levels of backward compatibility is required. Password policies do not require specifying any document recipients.

  Tip: Password security is unavailable if your application is operating in FIPS mode. Trying to save a document with password security applied results in an alert stating that this security method uses a non-FIPS compliant algorithm.

- **Public key certificate security**: Certificate security provides a high level of security, eliminates the need for password sharing, and allows assigning different permissions to different users whose identities can be verified and managed. Supported by Acrobat 6.0 and later.
- **Adobe LiveCycle Rights Management Server security**: These policies are stored on a server, and server access is required to use them. User access information is embedded in the document, so creating an ALCRMS policy involves specifying the document recipients from a list on the LiveCycle Server.

Table 2 Security method pros and cons

Method	Pros	Cons
Password	Backward-compatible to Acrobat 3.0 for certain encryption levels. Simple and easily understood. Share documents by sharing the password. Supports passwords for document opening. Supports password protecting document permission settings.	Protection depends on password strength. Anyone who knows the password has document access. All users share the same permissions. Won't work when the application is in FIPS mode.
Certificate	No password has to be remembered. Key is not susceptible to brute force discovery and resides only on the intended recipients machine. Can encrypt documents for specific people. Can use certificates issued by a trusted 3rd party certificate authority. Allows specifying different permission settings for users. Can leverage LDAP directories for recipients and group lists.	Users must have a digital ID. Organizations need to distribute and manage digital IDs. Compatible with Acrobat 5.0, but full support appears first in 6.0.
ALCRMS (server-based)	Centralized administration of security policies. Supports document auditing. Allows setting permissions for separate tasks such as opening, editing, and so on. Allows specifying different permission settings for users. Can leverage LDAP directories for recipients and group lists. Controls end-user offline access since authors can specify a validity time limit after which the document expires and is locked.	Requires a network connection. Requires an administrator and some infrastructure such as a LiveCycle Server.

2.1.2 Security Policies

Most workflows allow users to save the settings as a policy, thereby creating a reusable library of preconfigured security methods. When a policy author sets an encryption levels and recipient permissions and then saves them, the policy can later be applied to any document by choosing **Tools > Protection > Encrypt > Manage** and selecting a policy. Policies save time and ensure a consistently secure workflow. For details, see "Security Policies: Reusable Security Settings" on page 21.

2.1.3 Security Methods and Encryption

Encryption is used whenever a security method is added to a document. Security methods encrypt documents or parts of documents and are always involved in granting or denying permissions, thereby protecting content from unauthorized access and actions.

2.1.3.1 Encryption Workflow

The user workflow varies with the security method type as follows:

- **Password security**: The user is first asked to select a level of Acrobat backward compatibility. The selection automatically determines the encryption algorithm. Different document components can be encrypted based on the user's selection.

  Note: Password security is unavailable if your administrator has configured your application to operate in FIPS mode.

- **Certificate security**: The user selects what document components to encrypt and then chooses the encryption algorithm.
- **ALCRMS security**: The user selects what document components to encrypt. The algorithm is automatically applied by the server.

2.1.3.2 Choosing What to Encrypt

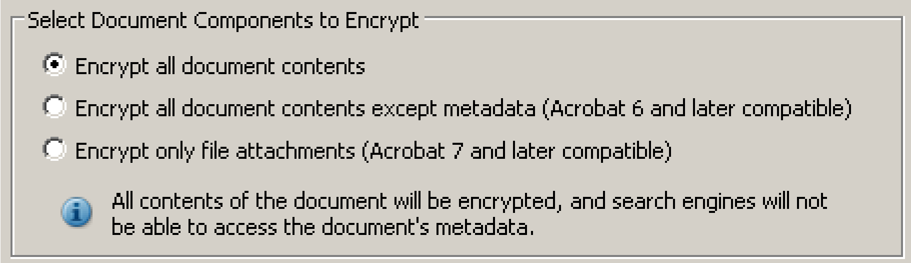
During the security method workflow, select a radio button in the **Select Document Components to Encrypt** panel to set the encryption options. You can encrypt all or part of a document based on your need for a specific security level and support for backward compatibility:

- **All contents**: Encrypts the document and its metadata (Acrobat 3 and later).
- **All contents except metadata**: Allows for document storage/retrieval systems and search engines to have access to the document metadata. A document open or a permissions password will required to access other document content.

  Encrypting everything except the metadata allows continued access to Acrobat's Catalog feature. By leaving the metadata unencrypted, users can catalog and index the metadata of encrypted documents, thereby making that data searchable (compatible with Acrobat 6 and later).

- **Only file attachments**: Allows full access to the document and encrypts only the file attachments. Permissions cannot be set on attachments. Using password security, a document open password is required for attachments (compatible Acrobat 7 and later).

Figure 7 Encryption configuration panel



Select Document Components to Encrypt

● Encrypt all document contents

○ Encrypt all document contents except metadata (Acrobat 6 and later compatible)

○ Encrypt only file attachments (Acrobat 7 and later compatible)

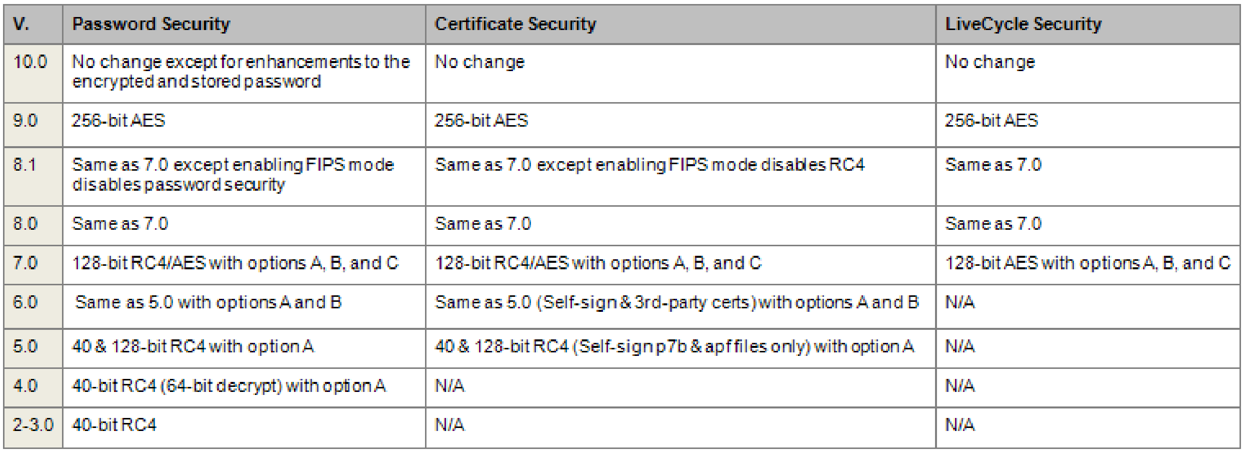
ℹ All contents of the document will be encrypted, and search engines will not be able to access the document's metadata.

2.1.3.3 Choosing an Algorithm and Compatibility Level

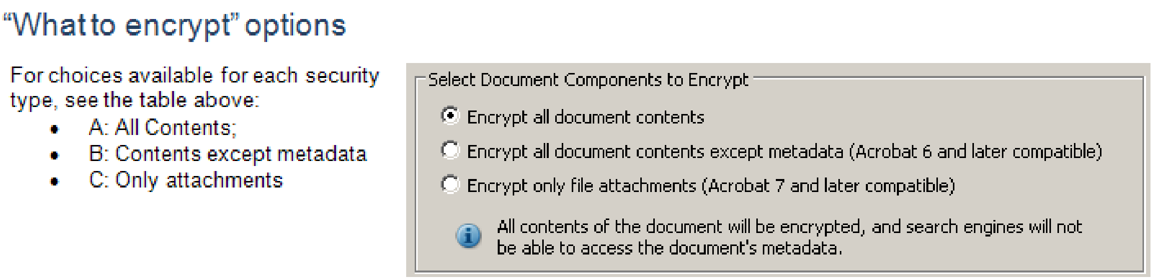
Acrobat's support for encryption algorithms evolves with each release. In general, selecting a more secure algorithm results in less backward compatibility.

Note: RC4 is unavailable if your administrator has configured your application to operate in FIPS mode.

Figure 8 Algorithms and compatibility



V.	Password Security	Certificate Security	LiveCycle Security
10.0	No change except for enhancements to the encrypted and stored password	No change	No change
9.0	256-bit AES	256-bit AES	256-bit AES
8.1	Same as 7.0 except enabling FIPS mode disables password security	Same as 7.0 except enabling FIPS mode disables RC4	Same as 7.0
8.0	Same as 7.0	Same as 7.0	Same as 7.0
7.0	128-bit RC4/AES with options A, B, and C	128-bit RC4/AES with options A, B, and C	128-bit AES with options A, B, and C
6.0	Same as 5.0 with options A and B	Same as 5.0 (Self-sign & 3rd-party certs) with options A and B	N/A
5.0	40 & 128-bit RC4 with option A	40 & 128-bit RC4 (Self-sign p7b & apf files only) with option A	N/A
4.0	40-bit RC4 (64-bit decrypt) with option A	N/A	N/A
2-3.0	40-bit RC4	N/A	N/A



"What to encrypt" options

For choices available for each security type, see the table above:

- A: All Contents;
- B: Contents except metadata
- C: Only attachments

Select Document Components to Encrypt

● Encrypt all document contents

○ Encrypt all document contents except metadata (Acrobat 6 and later compatible)

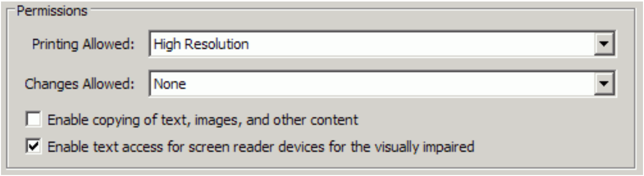
○ Encrypt only file attachments (Acrobat 7 and later compatible)

All contents of the document will be encrypted, and search engines will not be able to access the document's metadata.

2.1.4 Security Methods and Permissions

Permissions can be set whenever a security method is added to a document. Permission settings enable a document author to limit a document recipient's activities and interaction with a document. For example, restrictions can be placed on editing, copying, and printing. You set permissions by choosing the desired options in the Permissions panel when applying a security method.

Figure 9 Permissions panel



Permissions

Printing Allowed: High Resolution

Changes Allowed: None

□ Enable copying of text, images, and other content

☑ Enable text access for screen reader devices for the visually impaired

2.1.4.1 Permissions Workflow

The workflow varies slightly with the security method type as follows:

- **Password security**: The permissions panel appears at the beginning of the workflow. Checking **Restrict editing and printing of the document** enables all the other fields. Only the password security method requires a permissions password. If the document has a permission and a document open password, it can be opened with either password. The two passwords cannot be identical.

- **Certificate security**: The permissions panel appears at the end of the workflow. Permissions can be individually specified for different users by highlighting a specific recipient and choosing **Permissions**.
- **ALCRMS security**: The permissions are set ahead of time when the method is configured online. Permissions can be individually specified for different users by highlighting a document recipient and choosing **Permissions**. ALCRMS security provides the option of preventing a document recipient from saving and viewing the document offline, thereby storing a copy of the document on the local machine. This may not be desirable on public computers or when the computer is not secure.

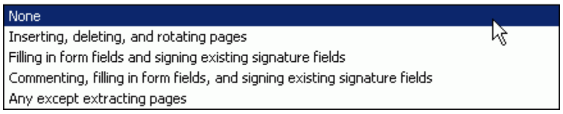
2.1.4.2 Permissions Options

All of the security methods provide the following options:

> **Note:** Adobe products enforce permissions restrictions. However, not all third-party products fully support and respect these permissions. Encryption and therefore document access would likely not be impaired, but Adobe cannot guarantee that individual permissions settings will remain function. Recipients using such third-party products might be able to bypass some of your restrictions.

Set the permissions as needed:

1. **Printing Allowed**:
   - **None**: Prohibits printing.
   - **Low Resolution**: Limits printing to 150-dpi resolution. Printing may be slower because each page is printed as a bitmapped image. This option is only available if a high encryption level (Acrobat 5 or Acrobat 6) is selected. Low resolution also inhibits users from printing a high quality document and using optical character recognition software to create a similar document with no security.
   - **High Resolution**: Allows printing at any resolution. For example, you can direct high-quality vector output to PostScript and other printers that support advanced high-quality printing features.
2. **Changes Allowed**: Limits page-level editing, commenting, and form field interaction.


None
Inserting, deleting, and rotating pages
Filling in form fields and signing existing signature fields
Commenting, filling in form fields, and signing existing signature fields
Any except extracting pages

   - **None**: Prevents users from changing the document, including filling in signature and form fields.
   - **Inserting, deleting, and rotating pages**: Lets users insert, delete, rotate pages, and create bookmarks and thumbnail pages. This option is only available if a high encryption level is selected.
   - **Filling in form fields and signing existing signature fields**: Lets users fill in forms and add digital signatures. This option doesn't allow users to add comments or create form fields.
   - **Commenting, filling in form fields, and signing existing signature fields**: Lets users fill in forms and add digital signatures and comments.

- **Any except extracting pages**: Lets users change the document using any method listed in the **Changes Allowed** menu, except remove pages.

3. **Enable copying of text, images, and other content**: Allows file contents (excluding comments) to be copied. It also makes the content available to assistive technology devices such as screen readers. It also lets utilities that need access to the contents of a PDF file, such as Acrobat Catalog, get to those contents. This option is only available if a high encryption level is selected.

4. **Enable text access for screen reader devices for the visually impaired**: Only available if the option above is NOT checked. Lets visually impaired users read the document with screen readers. This option doesn't allow users to copy or extract the document's contents. This option is only available if a high encryption level is selected.

2.1.5 Associating an Action with a Security Method

Acrobat can be configured to associate batch processes with a security method. When a batch process is associated with a security method, the method is invoked whenever a batch process is initiated.

To set the batch process security preference:

1. Choose **Edit > Preferences** (Windows) or **Acrobat > Preferences** (Macintosh).

2. Select **Action Wizard** in the left-hand tree.

3. Select a security method from the drop-down list.

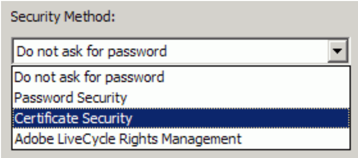
   The security handler does not apply security to files. Instead, it determines how batch processing deals with files that are password-protected.

   - If **Don't Ask For Password** is selected, the batch sequence proceeds as if the files are not secure.

   - If **Password Security** is selected, batch processing pauses when it encounters secured files and prompts for a password.

4. Choose **OK**.

Figure 10 Security methods for batch processing



Security Method: Do not ask for password

2.2 Changing and Viewing Security Settings

While anyone who can open a document can view its security methods, only those with permission can change those methods.

2.2.1 Viewing Document Encryption and Permission Settings

A document's security settings specify an encryption level (algorithm), what components are encrypted, and permissions. The document may be subject to additional restrictions if it is signed or certified. For more information, see "Viewing Document Restrictions" on page 19.

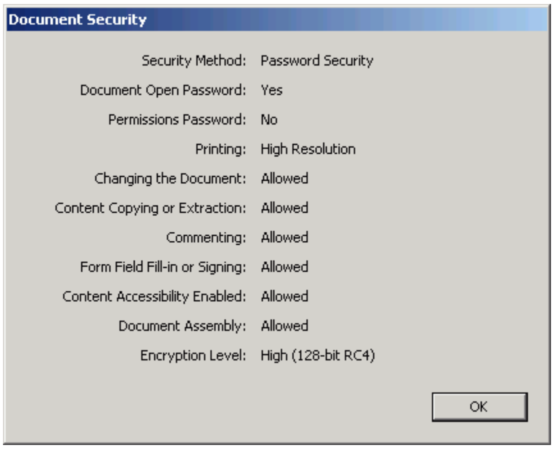
To view a document's encryption settings:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security** > **Show Security Properties**.

   Tip: You can also choose **File > Document Properties** or press **Control + D** and view the Security tab.

2. Choose **Show Details**. The settings are displayed in a dialog that varies with the security method type. The dialog does not update until a user saves and closes the document.

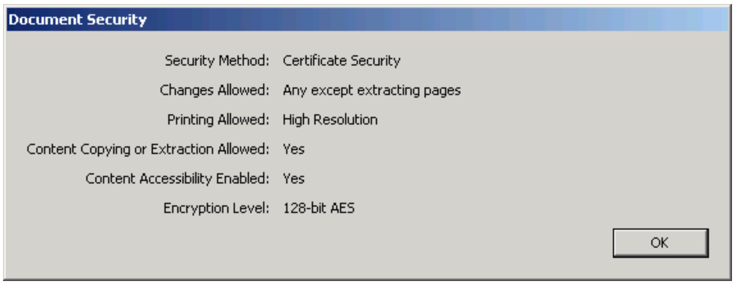
Figure 11 Document security settings: Password security



Document Security

Security Method: Password Security
Document Open Password: Yes
Permissions Password: No
Printing: High Resolution
Changing the Document: Allowed
Content Copying or Extraction: Allowed
Commenting: Allowed
Form Field Fill-in or Signing: Allowed
Content Accessibility Enabled: Allowed
Document Assembly: Allowed
Encryption Level: High (128-bit RC4)

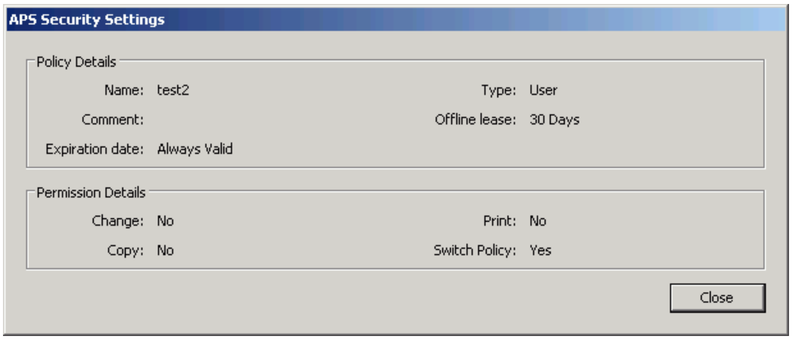
Figure 12 Document security settings: Certificate security



Document Security

Security Method: Certificate Security
Changes Allowed: Any except extracting pages
Printing Allowed: High Resolution
Content Copying or Extraction Allowed: Yes
Content Accessibility Enabled: Yes
Encryption Level: 128-bit AES

Figure 13 Document security settings: ALCRMS security



Name: test2
Type: User
Comment:
Offline lease: 30 Days
Expiration date: Always Valid

Change: No
Print: No
Copy: No
Switch Policy: Yes

2.2.2 Viewing Document Restrictions

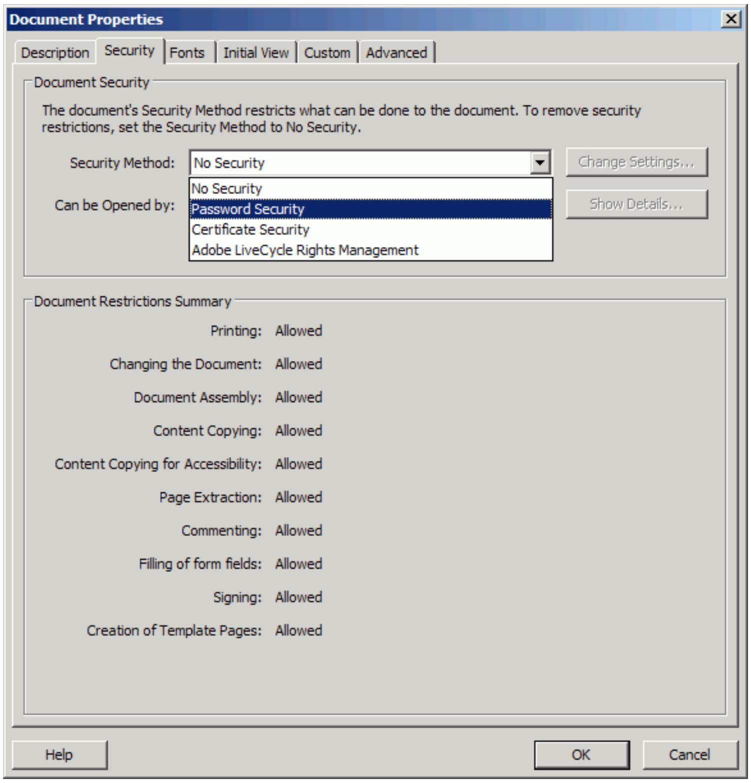
In addition to the encryption and permissions settings enforced by the document's security method, a document may be subject to additional restrictions if it is signed or certified. A summary of all security methods and signature-related restrictions appears in the Document Restrictions Summary panel.

When a document has restricted features, any tools and menu items related to those features are disabled. Users who are restricted from using certain document features they think they need should contact the document author.

To view the document restrictions summary in the Document Properties dialog, choose **Advanced** (Acrobat) or **Document** (Reader) **> Security > Show Security Properties.**

Tip: You can also choose **Control + D** and view the Security tab.

Figure 14 Document Property dialog



Document Security

The document's Security Method restricts what can be done to the document. To remove security restrictions, set the Security Method to No Security.

Security Method: No Security

- No Security
- Password Security
- Certificate Security
- Adobe LiveCycle Rights Management

Can be Opened by:

Document Restrictions Summary

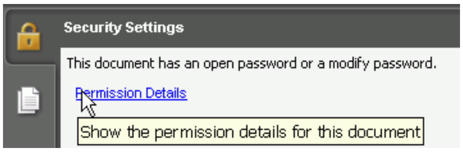
Printing:	Allowed
Changing the Document:	Allowed
Document Assembly:	Allowed
Content Copying:	Allowed
Content Copying for Accessibility:	Allowed
Page Extraction:	Allowed
Commenting:	Allowed
Filling of form fields:	Allowed
Signing:	Allowed
Creation of Template Pages:	Allowed

2.2.3 Viewing Security Settings in a Browser

To view document security settings in a Web browser:

1. Click on the lock icon in the left-hand pane.
2. Choose **Permissions Details**.

Figure 15 Security settings icon



Security Settings

This document has an open password or a modify password.

Permission Details

Show the permission details for this document

2.2.4 Changing the Security Method Type

Security settings cannot be changed on signed documents. Signature fields must first be cleared.

To change a document’s security method:

1. Choose **File > Properties > Security tab**.
2. Choose a new security method from the drop-down list.

Note: New settings take affect after the document is closed and reopened..

3. If the document is password protected, enter the document password.
4. Choose a security method and configure it. For details, see the following:
   - Chapter 3, "Password Security"
   - Chapter 4, "Certificate Security"
   - Chapter 5, "LiveCycle Rights Management Server Security"
5. Choose **OK**.
6. Save the document. New settings take affect after the document is closed and reopened..

2.2.5 Editing Security Method Settings

To change the security settings for an encrypted document:

1. Choose **File > Properties > Security tab**.
2. In the Security panel, choose **Change Settings**. For details, see the following:
   - Chapter 3, "Password Security"
   - Chapter 4, "Certificate Security"
   - Chapter 5, "LiveCycle Rights Management Server Security"

   Note: New settings take affect after the document is closed and reopened..
3. Save the document.

2.2.6 Removing Document Security

Security can be removed from an open document by those with permissions to do so. You may be required to enter a password or have the requisite certificate to remove a policy.

To remove security settings from a document:

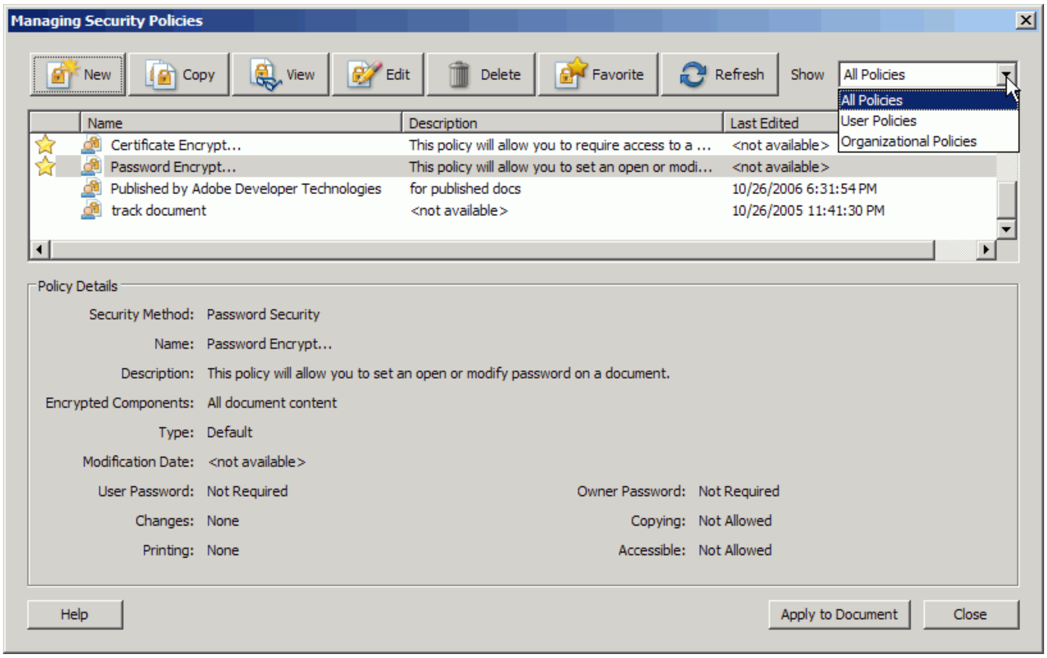
1. From the toolbar, choose **File > Properties > Security tab** > **No Security**.
2. If prompted, type the permissions password.
3. When asked to confirm removal of the security settings, choose **OK**.
4. Choose **OK**. **Tools > Protection >**

2.3 Security Policies: Reusable Security Settings

Security policies provide a way to save and reuse security method settings. Saving your configured security method as a policy saves time and effort later. Policies are not embedded in a document. When a document is sent to someone, it contains the specified security settings, but the policy stays with the policy author.

Note: Policies can be applied to documents created with any version of Acrobat. However, specific policy settings may not be supported for documents created with some earlier versions.

Figure 16 Security method selection from Policy Manager



	Name	Description	Last Edited
☆	Certificate Encrypt...	This policy will allow you to require access to a ...	<not available>
☆	Password Encrypt...	This policy will allow you to set an open or modi...	<not available>
	Published by Adobe Developer Technologies	for published docs	10/26/2006 6:31:54 PM
	track document	<not available>	10/26/2005 11:41:30 PM

Policy Details

Security Method: Password Security
Name: Password Encrypt...
Description: This policy will allow you to set an open or modify password on a document.
Encrypted Components: All document content
Type: Default
Modification Date: <not available>
User Password: Not Required
Owner Password: Not Required
Changes: None
Copying: Not Allowed
Printing: None
Accessible: Not Allowed

Policy settings include two main kinds of information:

- Encryption type, permission settings, and passwords (if any).
- Information about the individuals or groups that can open a document or change its security settings.

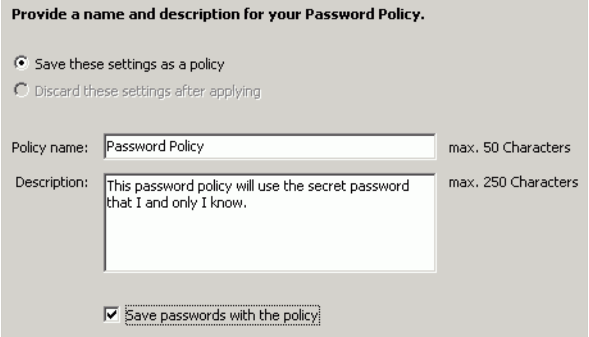
There are two categories of security policy sources. These policies can be displayed together or individually (Figure 16):

- **User policies**: User policies are created and applied by anyone. User password and certificate policies are stored locally while Adobe LiveCycle Rights Management Server policies are stored on the server. Policy authors can edit and delete the policies they create.
- **Organizational policies**: An organizational policy is created by an Adobe LiveCycle Rights Management Server administrator and is stored on a policy server. The server controls access to documents and auditing events as defined by the security policy. Only a policy administrator can edit, and delete organizational policies.

2.3.1 Creating Security Policies with Policy Manager

Policies can be created ahead of time or during the course of creating new security settings. When the Security Settings Console appears, simply choose **Save these settings as a policy** and enter a policy name and optional description (Figure 17).

Figure 17 Security policy: General settings



Provide a name and description for your Password Policy.

● Save these settings as a policy

○ Discard these settings after applying

Policy name: Password Policy — max. 50 Characters

Description: This password policy will use the secret password that I and only I know. — max. 250 Characters

☑ Save passwords with the policy

To create a security policy ahead of time with Policy Manager:

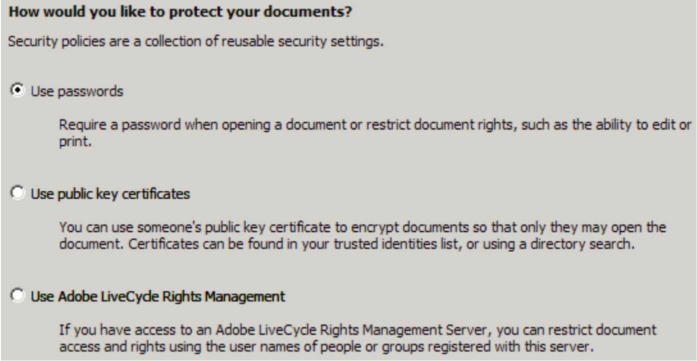
1. Choose **Tools** > **Protection** > **Encrypt** > **Manage** (Figure 16).

2. Choose **New**.

3. Select a security method for the policy and configure in the appropriate section:

- Chapter 3, “Password Security”
- Chapter 4, “Certificate Security”
- Chapter 5, “LiveCycle Rights Management Server Security”

Figure 18 Policy security method selection



How would you like to protect your documents?

Security policies are a collection of reusable security settings.

● Use passwords

Require a password when opening a document or restrict document rights, such as the ability to edit or print.

○ Use public key certificates

You can use someone's public key certificate to encrypt documents so that only they may open the document. Certificates can be found in your trusted identities list, or using a directory search.

○ Use Adobe LiveCycle Rights Management

If you have access to an Adobe LiveCycle Rights Management Server, you can restrict document access and rights using the user names of people or groups registered with this server.

2.3.2 Applying a Security Policy to a Document

Organization and user policies can be applied to any document by those who have permission to do so.

To apply a security policy to a document:

1. Choose **Tools** > **Protection** > **Encrypt** > **Manager**.

2. Highlight a policy.

3. Choose **Apply to Document**.

4. Save the document.

Tip: If a policy has been designated as a "favorite," a star appears next to the selected policy. All favorites appear in the security menu (Figure 19).

2.3.3 Viewing a Security Policy

To view a security policy:

1. Choose **Tools** > **Protection** > **Encrypt** > **Manage** (Figure 16).
2. Choose a security policy.
3. Choose **View**.

   The policy opens in read-only mode and cannot be edited.

2.3.4 Copying a Security Policy

Copying a policy is useful when a new policy is needed that is similar to an existing policy. The first policy is simply copied, edited, and saved under a new name.

To copy a security policy:

1. Choose **Tools** > **Protection** > **Encrypt** > **Manage**(Figure 16).
2. Choose a security policy.
3. Choose **Copy**.
4. Change the policy's settings as described in one of the following sections:
   - Chapter 3, "Password Security"
   - Chapter 4, "Certificate Security"
   - Chapter 5, "LiveCycle Rights Management Server Security"

2.3.5 Editing a Security Policy

Existing policies can be edited. For example, if a document is distributed to a group of users and the owners wants to revoke permission for others to open it, the owner can change the policy.

To edit a security policy:

1. Choose **Tools** > **Protection** > **Encrypt** > **Manage** (Figure 16).
2. Choose a security policy.
3. Choose **Edit**.
4. Change the policy's settings as described in one of the following sections:
   - Chapter 3, "Password Security"
   - Chapter 4, "Certificate Security"
   - Chapter 5, "LiveCycle Rights Management Server Security"

2.3.6 Making a Security Policy Favorite

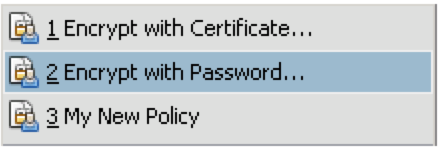
When a policy is selected as a favorite, a star appears next to that policy and the policy is then listed on the security menu. Any new policy you create is automatically made a favorite.

To make a security policy favorite:

1. Choose **Tools** > **Protection** > **Encrypt** > **Manage** (Figure 16).
2. Choose a security policy.
3. Choose **Favorite**.
4. Choose **Close**.

   A star appears next to the selected policy. "Favorited" policies appear in the security menu (Figure 19).

Figure 19 Security policy: Favorites list



2.3.7 Refreshing the Security Policy List

If the policies are available via a server, refresh the security policy list to ensure that you have access to the most up-to-date server policies.

1. Choose **Tools** > **Protection** > **Encrypt** > **Manage**.
2. Choose **Refresh**.
3. When the login screen appears, enter a username and password.
4. Choose **OK**.

2.3.8 Deleting a Security Policy

A user can delete any policy that they created. It is not possible to delete organizational policies created by an administrator.

To delete a security policy:

1. Choose **Tools** > **Protection** > **Encrypt** > **Manage** (Figure 16).
2. Choose a security policy.
3. Choose **Delete**.
4. Choose **Yes** at the confirmation dialog.

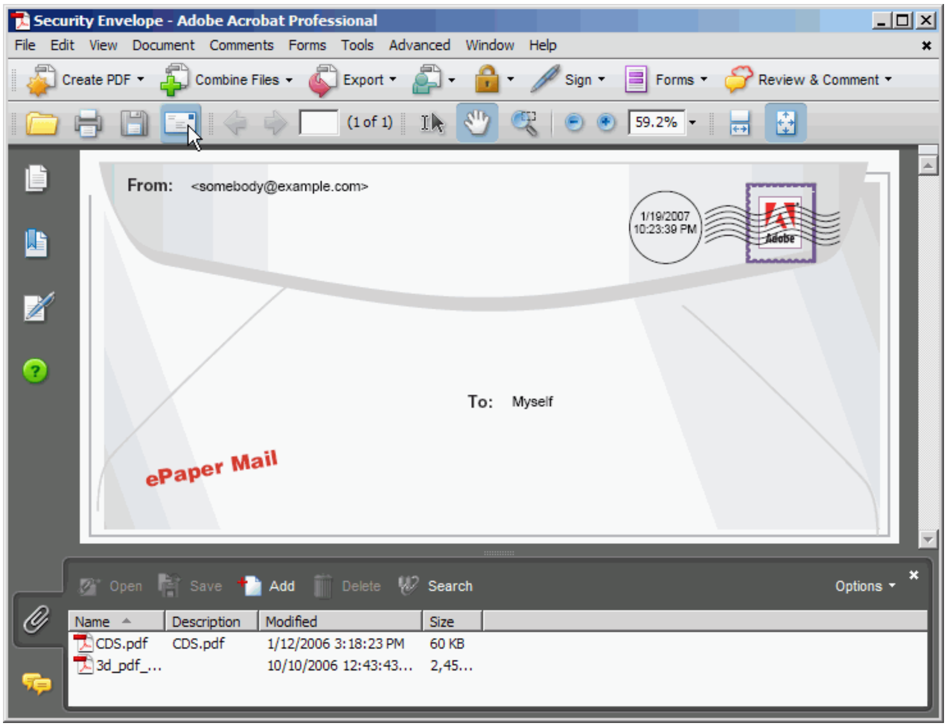
5. Choose **Close**.

2.4 Envelopes

You can add security to one or more documents by embedding them in an encrypted envelope, called a security envelope. Envelopes are simply PDF files with attachments. This method is especially useful if you want to send a secure file attachment without modifying or encrypting the attached files When someone opens the envelope, they can extract the attachments and save them to disk. The saved files are identical to the original file attachments and are no longer encrypted when saved.

For example, suppose that you want to send several documents, including non-PDF documents, to your accountant, but you don't want anyone else to view the documents. You can embed these documents as file attachments in a security envelope, encrypt the security envelope so that only your accountant can open the attachments, and then email it. Anyone can open the envelope, view its cover page, and even view a list of the contents of that envelope, but only your accountant can view the embedded attachments and extract them to the computer.

Figure 20 Security envelope



From: <somebody@example.com>

To: Myself

Name	Description	Modified	Size
CDS.pdf	CDS.pdf	1/12/2006 3:18:23 PM	60 KB
3d_pdf_...		10/10/2006 12:43:43...	2,45...

Embed file attachments in security envelopes for secure transit.

1. Choose the **Tools > Protection > More Protection >Create Security Envelope**.

2. Choose **Add File To Send**.

3. Browse to the documents you want to attach and choose **Open**.

   Select any PDFs in the list that you don't want to include and choose **Remove Selected Files**.

4. Choose **Next**.

5. Select an envelope template.

6. Choose **Next**.

7. Select whether to deliver the envelope now or later. In most cases, you will want to choose **Send the envelope later** so you can view it and fill out its form fields before sending.

   Note: Templates sometimes contain form fields (such as **To** and **From**) that you can fill in before sending. If you choose to send now and a dialog asks if you really want to send before filling in these fields, choose **Yes** or **No** to continue.

8. Choose **Next**.

9. Choose **Next** OR apply a security policy. Security policies are optional.

   Select **Show All Policies**, and then select a security policy from the list of available policies (or create a new policy if needed).

   Tip: Follow the on-screen instructions to complete the security envelope. If prompted, provide your identity information.

10. Choose **Finish**.

11. Type an email address in the message that appears and choose **Send**, or save the security envelope to send later.

3 Password Security

Acrobat users can perform any task in this section. Adobe Reader users can only view encrypted documents and can not encrypt them for others.

Password security provides a simple method for sharing encrypted documents by sharing passwords. Like all security methods, password security can enforce document restrictions on operations such as opening, printing, and editing. Since password security does not provide the ability to specify different permissions for different users, everyone that can open the document will have the same permissions.

Document protection has a dependency on password strength. Acrobat 9 .0 now allows full Unicode pass phrases up to 128 characters in length (an actual limit of 128 UTF-8 bytes). Acrobat 8.x and earlier limits passwords to 32 characters maximum and almost entirely to the Latin alphabet (strictly, PDFDocEncoding). Password security encryption levels may also be set to be backward-compatible to Acrobat 3.0.

Note: Password security is unavailable if your administrator has configured your application to operate in FIPS mode.

Password security also provides separate options for opening the document and setting user permissions; therefore, password security uses two kinds of passwords: a Document Open password and a Permissions password.

- **Document open password**: Required to open a password-protected document.
- **Permissions password**: Required to change permissions such as those for copying and editing.

Tip: If the document has both types of passwords, it can be opened with either password. The document open password and permissions password cannot be identical.

Note: A Reader user can use the permissions password to open the document, but they will not be able to change permissions.

At a high level, adding password security includes specifying encryption settings, creating a Document Open password (if needed), creating a Permissions password (if needed), specifying permissions settings, and saving the document.

3.1 Creating Password Security Settings

Configure and add password security to a document by either creating a policy which can be saved and reused or by creating them once and discarding them. For details, see:

- "Creating a Reusable Password Security Policy" on page 28
- "Creating Password Security for One-Time Use" on page 31

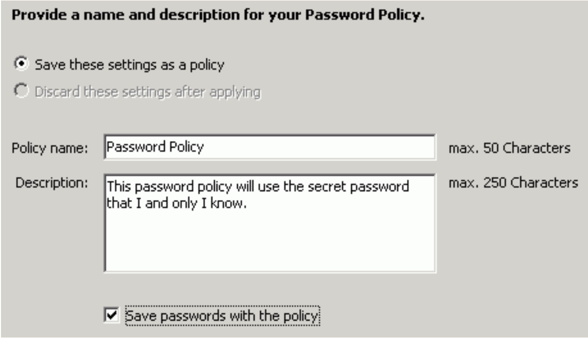
3.1.1 Creating a Reusable Password Security Policy

1. Choose **Tools** > **Protection** > **Encrypt** > **Manage** to open the Policy Manager.

2. Choose **New**.
3. Choose **Use passwords**.
4. Choose **Next**.
5. Enter a policy name and optional description.
6. Check or uncheck **Save passwords with the policy**.

   Tip: You can save the password with the policy so that it's automatically used, or you can have Acrobat prompt you for the policy each time you apply it.

   Figure 21 Security policy: General settings

   Provide a name and description for your Password Policy.

   ● Save these settings as a policy
   ○ Discard these settings after applying

   Policy name: Password Policy — max. 50 Characters

   Description: This password policy will use the secret password that I and only I know. — max. 250 Characters

   ☑ Save passwords with the policy

7. Choose **Next**.
8. Configure the security settings dialog:
   1. **Compatibility**: The compatibility options determine what encryption options will be available. Compatibility with earlier versions of Acrobat may mean all document contents will have to be encrypted. Set the compatibility level as follows:
      - **Acrobat 3.0 and later**: Encryption uses the 40-bit RC4 encryption algorithm. This setting forces the encryption of strings and streams only and limits other features.
      - **Acrobat 5.0 and later**: Encryption uses the 128-bit RC4 encryption algorithm. This setting allows the accessibility option to be selected independently of the copy option, restricts printing to 150-bit dpi, and expands the set of **Changes Allowed** options.
      - **Acrobat 6.0 and later**: Encryption uses the 128-bit RC4 algorithm. This setting allows the option of leaving the document metadata unencrypted while the remainder of the document is encrypted. All of the options for Acrobat 5.0 and later are also available.
      - **Acrobat 7.0 and later**: Encryption uses the 128-bit AES algorithm. When selected, the option of only encrypting the file attachments is available as well as all of the previous options.
      - **Acrobat 9.0 and later**: Encryption uses the 256-bit AES algorithm. Password length can be up to 64 characters.

      Note: The Acrobat 9.0 and later option is removed in Acrobat X.

      - **Acrobat X and later**: Encryption uses the 256-bit AES algorithm. Password length can be up to 64 characters.

    2. Configure the **Select Document Components to Encrypt** panel as described in "Choosing What to Encrypt" on page 14

9. If you would like to control who can open the document, provide a Document Open password. You must provide a document open, a permissions password, or both. If you only need to create a permissions password, skip to Step 8.

    - Check **Require a password to open the document**.
    - Enter a password.

10. If you would like to use password-based permissions, check **Use permissions password to restrict editing of security settings**. Otherwise, skip to Step 12.

    Document authors can set a permissions password that allows users to change the document's permissions. Only a holder of the permissions password will be able to change the permissions. The Permission password can also be used to open the document even if there is a separate Document Open password.

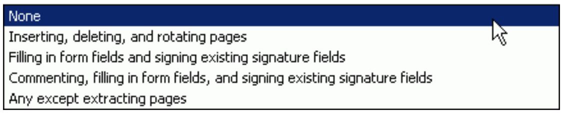
    Tip: Adobe recommends that permission passwords and document open password always be used together. The permissions password is used to change permissions and is NOT needed to gain access to the features the author is permitting. Thus, holders of the permissions password are essentially "owners" of the document and can do anything to it that the author could do.

    Caution: Adobe products enforce permissions restrictions. However, not all third-party products fully support and respect these permissions (the encryption would not be violated). Recipients using such third-party products might be able to bypass some of your restrictions.

    Set the permissions as needed:

    1. **Printing Allowed**:
        - **None**: Prohibits printing.
        - **Low Resolution**: Limits printing to 150-dpi resolution. Printing may be slower because each page is printed as a bitmapped image. This option is only available if a high encryption level (Acrobat 5 or Acrobat 6) is selected. Low resolution also inhibits users from printing a high quality document and using optical character recognition software to create a similar document with no security.
        - **High Resolution**: Allows printing at any resolution. For example, you can direct high-quality vector output to PostScript and other printers that support advanced high-quality printing features.

    2. **Changes Allowed**: Limits page-level editing, commenting, and form field interaction.



None
Inserting, deleting, and rotating pages
Filling in form fields and signing existing signature fields
Commenting, filling in form fields, and signing existing signature fields
Any except extracting pages

        - **None**: Prevents users from changing the document, including filling in signature and form fields.

- **Inserting, deleting, and rotating pages**: Lets users insert, delete, rotate pages, and create bookmarks and thumbnail pages. This option is only available if a high encryption level is selected.
- **Filling in form fields and signing existing signature fields**: Lets users fill in forms and add digital signatures. This option doesn't allow users to add comments or create form fields.
- **Commenting, filling in form fields, and signing existing signature fields**: Lets users fill in forms and add digital signatures and comments.
- **Any except extracting pages**: Lets users change the document using any method listed in the **Changes Allowed** menu, except remove pages.

3. **Enable copying of text, images, and other content**: Allows file contents (excluding comments) to be copied. It also makes the content available to assistive technology devices such as screen readers. It also lets utilities that need access to the contents of a PDF file, such as Acrobat Catalog, get to those contents. This option is only available if a high encryption level is selected.

4. **Enable text access for screen reader devices for the visually impaired**: Only available if the option above is NOT checked. Lets visually impaired users read the document with screen readers. This option doesn't allow users to copy or extract the document's contents. This option is only available if a high encryption level is selected.

11. Choose **OK**.

12. Reenter the Document Open and/or Permissions passwords (if any) when asked to confirm it and choose **OK**.

13. Choose **Finish**.

3.1.2 Creating Password Security for One-Time Use

Use this method if you:

- Need to make the document backward-compatible to Acrobat 3.0.
- Do not need to save the settings as a policy.

To apply password security to the current document:

1. Choose **File > Properties > Security tab.**

2. Select **Password Security** from the Security Method menu.

3. Check or uncheck **Save passwords with the policy**.

4. Choose **OK**.

5. Set the compatibility level to control what encryption options will be available. Compatibility with earlier versions of Acrobat may require encrypting all document contents. Compatibility levels include:

   - **Acrobat 3.0 and later**: All document contents are encrypted with the 40-bit RC4 algorithm. This option provides the most limited set of permission setting options.

- **Acrobat 5.0 and later**: Encryption uses the 128-bit RC4 encryption algorithm. This setting allows the accessibility option to be selected independently of the copy option, restricts printing to 150-bit dpi, and expands the set of **Changes Allowed** options.
- **Acrobat 6.0 and later**: Encryption uses the 128-bit RC4 algorithm. This setting allows the option of leaving the document metadata unencrypted while the remainder of the document is encrypted. All of the options for Acrobat 5.0 and later are also available.
- **Acrobat 7.0 and later**: Encryption uses the 128-bit AES algorithm. When selected, the option of only encrypting the file attachments is available as well as all of the previous options.
- **Acrobat 9.0 and later**: Encryption uses the 256-bit AES algorithm. When selected, the option of only encrypting the file attachments is available as well as all of the previous options.

6. Configure the **Select Document Components to Encrypt** panel as described in "Choosing What to Encrypt" on page 14.

7. If you would like to control who can open the document, provide a Document Open password. You must provide a document open, a permissions password, or both. If you only need to create a permissions password, skip to Step 8.

    1. Check **Require a password to open the document**.

    2. Enter a password.

8. If you would like to use password-based permissions, check **Restrict editing and printing of the document**. Otherwise, skip to Step 11.

    Document authors can set a permissions password that allows users to change the document's permissions. Only a holder of the permissions password will be able to change the permissions. Permission password can also open the document even if there is a separate Document Open password.

    Tip: Adobe recommends that permission passwords and document open password always be used together. The permissions password is used to change permissions and is NOT needed to gain access to the features the author is permitting. Thus, holders of the permissions password are essentially "owners" of the document and can do anything to it that the author could do.

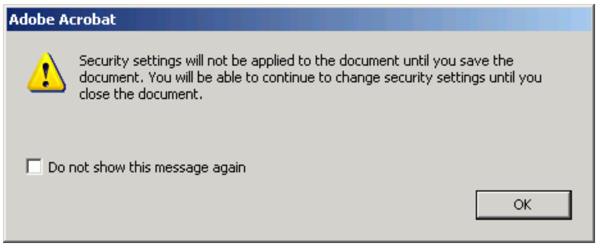
9. Set the permissions as described in "Permissions Options" on page 16.

10. Choose **OK**.

11. Reenter the Document Open and/or Permissions passwords (if any) when asked to confirm it and choose **OK**.

12. If an alert appears indicating the changes won't be applied until the document is saved, choose **OK**.

Figure 22 Security settings require "save" alert



Adobe Acrobat

Security settings will not be applied to the document until you save the document. You will be able to continue to change security settings until you close the document.

☐ Do not show this message again

13. Save the document. New settings take affect after the document is closed and reopened..

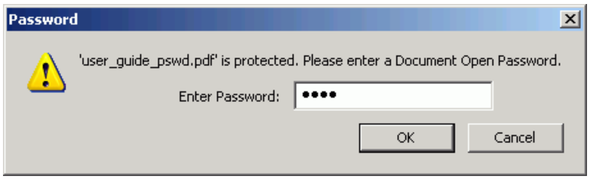
3.2 Opening a Password-Protected Document

You must know the Document Open or Permissions password to open the document.

To open a password protected document:

1. Open the document.
2. Enter the password.
3. Choose **OK**.

Figure 23 Password prompt



Password

'user_guide_pswd.pdf' is protected. Please enter a Document Open Password.

Enter Password: ••••

3.3 Removing Password Security

To remove password security settings from a document:

1. Open the document and supply the required password.
2. Choose **File > Properties > Security tab > No Security**.
3. When asked to confirm that you would like to remove security, choose **OK**.
4. Save the document to have the change take effect.

3.4 Password Recovery

Caution: There is no way to recover a lost password from a document. Keep a backup copy that is not password-protected.

4 Certificate Security

Acrobat users can perform any task in this section. Adobe Reader users can only view encrypted documents and not encrypt them for others.

If you share documents that require high security, you may need certificate security. Businesses use certificate security because a public key infrastructure (PKI) enables central management by an administrator. The administrator can set up an LDAP directory server for providing certificate access, create custom certificates for specialized workflows, and so on. Where secure PDFs do not have to be compatible with Acrobat versions prior to 6.0, certificate security has several advantages:

- **Different users can have different permission settings**: Unlike password-based security which applies permissions equally to everyone, certificate security allows authors to specify permissions for individuals. For example, it is possible to give employees the ability to sign documents and fill in form fields while giving only managers the ability to add comments or delete pages. Permissions are useful for distributing documents to users that need varied document access and usage rights.
- **Superior security attributes**: No password has to be remembered or shared as the public and private keys to encrypt and decrypt documents reside only on the machines of those participating in secure workflows. These keys are less susceptible to brute force discovery than passwords.

Note: Participants in a certificate security workflow must have a digital ID and cannot use Acrobat versions prior to 6.0.

4.1 Setting up the Certificate Security Environment

If you're going to use certificate security, consider doing the following:

- Configuring Acrobat to use certificates in the Windows Certificate store as well as those in the Acrobat store (which is on by default).
- Choosing which certificate to use for encryption for those contacts who have provided you with more than one.
- Setting up a group and a reusable security policy to simplify your workflows.

4.1.1 Accessing the Windows Certificate Store

The Windows Certificate Store contains certificates used by Windows applications. For example, when signing outgoing emails in Outlook, the digital ID comes from the Personal certificate store in Windows. The trusted certificates stored in the Windows Trusted People certificate store are used by Windows applications to validate signed emails from other people.

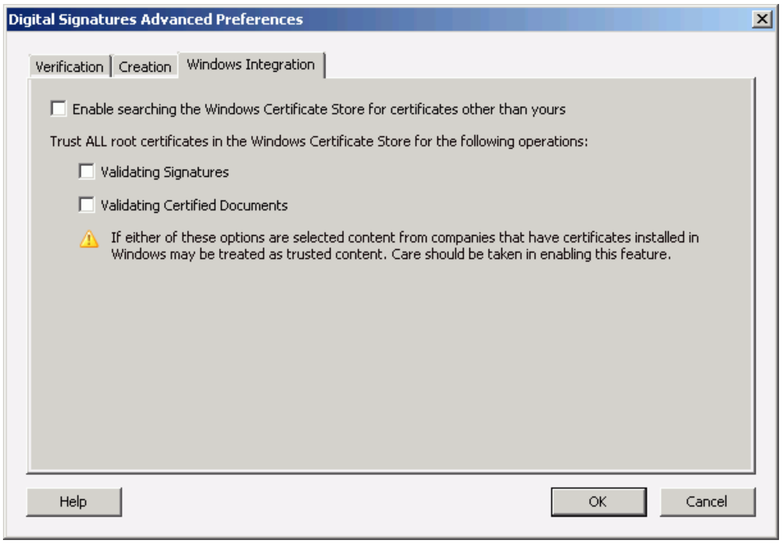
If you want to use certificates in the Windows Certificate Store to encrypt the document for the certificate owner, add the Windows store to the certificate search path. This allows you to search Windows directories when applying certificate security. By default, the Windows Certificate Store is not included in the application search path. Once the option is manually turned on, the Windows store will appear in the Search for recipients dialog **Directories** drop-down list.

To enable searching for certificates in the Windows Certificate Store:

1. Choose **Edit > Preferences**.
2. Choose **Security** in the left-hand panel.
3. Choose **Advanced Preferences**.
4. Display the Windows Integration tab.
5. Check **Enable searching the Windows Certificate Store for certificates other than yours**.
6. Choose **OK**.
7. Choose **OK**.

Tip: The checkboxes related to trust are only used for signature validation.

Figure 24 Windows integration



☐ Enable searching the Windows Certificate Store for certificates other than yours

Trust ALL root certificates in the Windows Certificate Store for the following operations:

☐ Validating Signatures

☐ Validating Certified Documents

⚠ If either of these options are selected content from companies that have certificates installed in Windows may be treated as trusted content. Care should be taken in enabling this feature.

The Windows Certificate Store will now appear in Search for Recipients dialog's directory list. The dialog can be invoked from two locations:

- From a certificate security workflow: Set the encryption settings, choose **Next**, and then choose **Search**.
- From the Trusted Identity Manager: Choose **Add Contacts**, and then choose **Search**.

4.1.2 Selecting a Certificate to Use for Encryption

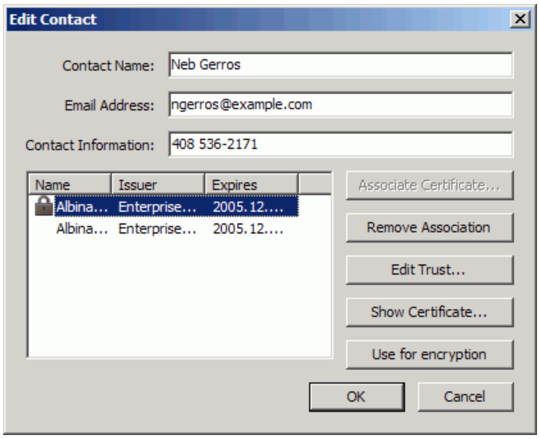
Because you encrypt a document for someone with the public key in their certificate, you must first explicitly choose their certificate for encryption. Each contact in your Trusted Identity list should be associated with at least one certificate. If there is only one certificate, Acrobat automatically selects it as the one to use for encryption. If more than one certificate is associated with the contact, you can select which one to use as the default encryption certificate.

Note: To use a certificate for encryption must have encryption usage rights. A warning dialog appears during the encryption process if the selected certificate cannot be used.

To set a default certificate for encryption:

1. Choose **Advanced > Manage Trusted Identities**.
2. Choose a contact in the left-hand list.
3. Choose **Details**.
4. Highlight a certificate in the certificate list.
5. Choose **Use for encryption** (Figure 25). The lock icon moves to the selected certificate.
6. Choose **OK**.

Figure 25 Choosing a certificate for encryption



Edit Contact

Contact Name: Neb Gerros

Email Address: ngerros@example.com

Contact Information: 408 536-2171

Name	Issuer	Expires
Albina...	Enterprise...	2005.12....
Albina...	Enterprise...	2005.12....

4.2 Working with Groups of Contacts

Contacts can be added to a group so that all group members can easily share a predefined set of permissions and restrictions. For example, it is possible to create a certificate-based security policy that applies to an entire group. Administrators and home users can create a group and export the group's details to an FDF file that is then sent to individual users. This feature makes it easy to manage permissions for a large number of people.

Note: Importing a group imports the contacts (all group members), but not the group. If desired, create a new group from those newly imported contacts.

4.2.1 Creating a Group

Individual users and administrators create a group using the same method.

To create a group:

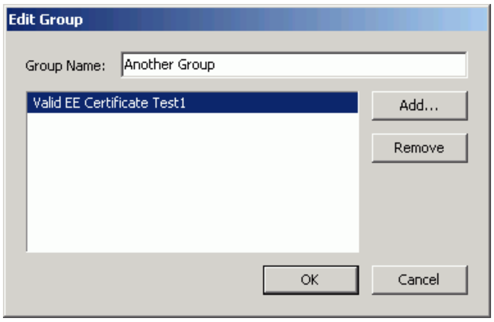
1. Choose **Advanced** > **Manage Trusted Identities**.
2. Choose **New Group**.
3. Enter a group name (Figure 26).
4. Add contacts .
5. Choose **OK**.

4.2.2 Adding or Removing Group Contacts

To add or remove group members:

1. Choose **Advanced** > **Manage Trusted Identities**.
2. Double-click on a group or highlight the group and choose **Details**.
3. Add or remove a contact:
   - **Adding a contact**: Choose **Add**, select a contact from the contact list, and choose **OK** twice.
   - **Removing a contact**: Select a contact, choose **Remove**, and choose **OK**.

Figure 26 Contacts: Editing a group



Edit Group

Group Name: Another Group

Valid EE Certificate Test1

4.3 Creating Certificate Security Settings

When adding security to a document, you either create a policy which can be reused or creating the once and discard them. For details, see:

- "Creating a Reusable Certificate Security Policy" on page 38
- "Creating Certificate Security for the Current Document" on page 42

4.3.1 Creating a Reusable Certificate Security Policy

To create a certificate security policy:

1. Choose **Tools** > **Protection** > **Encrypt** > **Manage**.

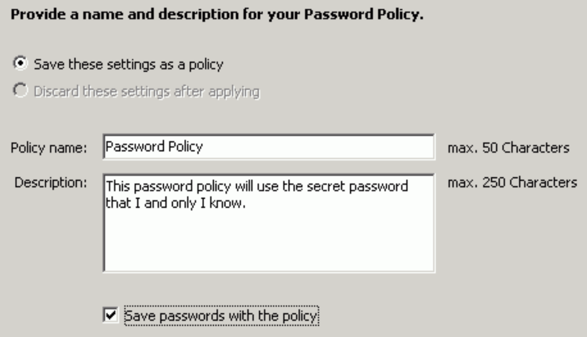
2. Choose **New**.

3. Select **Use public key certificates**.

4. Choose **Next**.

5. Enter a policy name and optional description.

Figure 27 Security policy: General settings



Provide a name and description for your Password Policy.

● Save these settings as a policy

○ Discard these settings after applying

Policy name: Password Policy max. 50 Characters

Description: This password policy will use the secret password that I and only I know. max. 250 Characters

☑ Save passwords with the policy

6. Configure the **Select Document Components to Encrypt** panel as described in "Choosing What to Encrypt" on page 14.

7. Check or uncheck **Ask for recipients when applying this policy**.
   - If checked, you will not be asked in the next step to select the recipient certificates. Because the policy will not be associated with any recipients you will select them when you apply the policy.
   - If unchecked, you will be asked to select certificates now so that the document recipients will be identified in the policy.

8. Choose the encryption algorithm:
   - **128-bit RC4**: Compatible with Acrobat 6.0 and later as well as other non-Adobe and Adobe PDF clients such as Ghostscript and Apple Preview that have not implemented AES. RC4 has a smaller file size by about 32 bytes per stream.
   - **128-bit AES:** Compatible with Acrobat 7.0 and later. It is mandated for some U.S. government documents because it is more secure than RC4. AES adds up to 32 bytes per stream.
   - **256-bit AES**: Compatible with Acrobat 9.0 and later. Provides the highest level of encryption.

   Note: The RC4 encryption algorithm is unavailable if your administrator has configured your application to operate in FIPS mode.

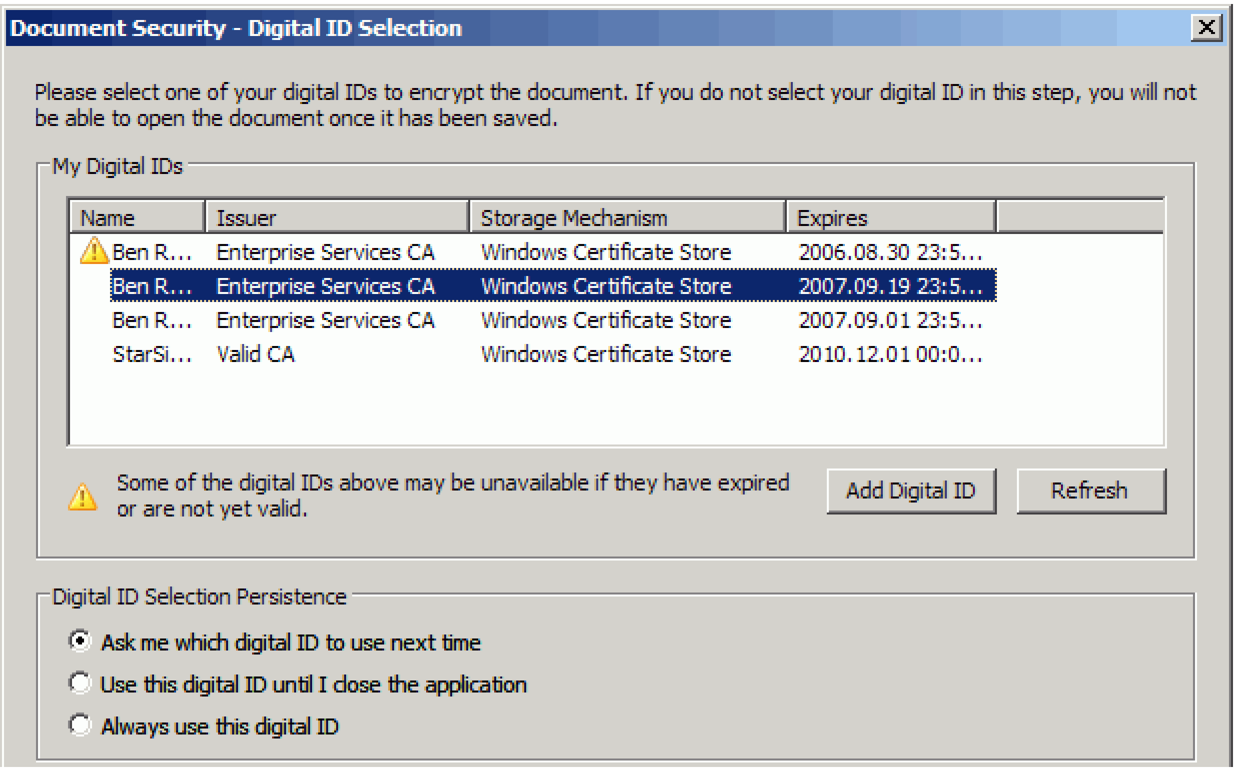
9. Choose **Next**. If you checked **Ask for recipients when applying this policy**, choose **Finish**. Otherwise, go to the next step.

10. *The Digital ID Selection dialog may not appear. If it does not appear, go to the "add document recipients to the recipient list" step*: The digital ID selection dialog only appears if you have no digital IDs suitable for encryption or more than one. If you only have one digital ID suitable for encryption, then this dialog does not appear (for example, one ID is set as the default for encryption in the Security Settings Console). If the dialog does appear, select your digital ID that you will use to access this document in the future.

Tip: While it is possible to apply certificate security without selecting your digital ID, doing so leaves you off of the recipient list and permanently locks you out of the document.

If the required digital ID does not appear in the list, choose **Add Digital ID**.

Figure 28 Choosing a digital ID for certificate security



Document Security - Digital ID Selection

Please select one of your digital IDs to encrypt the document. If you do not select your digital ID in this step, you will not be able to open the document once it has been saved.

My Digital IDs

Name	Issuer	Storage Mechanism	Expires
⚠ Ben R...	Enterprise Services CA	Windows Certificate Store	2006.08.30 23:5...
Ben R...	Enterprise Services CA	Windows Certificate Store	2007.09.19 23:5...
Ben R...	Enterprise Services CA	Windows Certificate Store	2007.09.01 23:5...
StarSi...	Valid CA	Windows Certificate Store	2010.12.01 00:0...

⚠ Some of the digital IDs above may be unavailable if they have expired or are not yet valid.

Digital ID Selection Persistence

● Ask me which digital ID to use next time
○ Use this digital ID until I close the application
○ Always use this digital ID

11. If you have more than one digital ID, choose the digital ID persistence level.
    - Ask me which digital ID to use next time
    - Use this digital ID until I close the application
    - Always use this digital ID

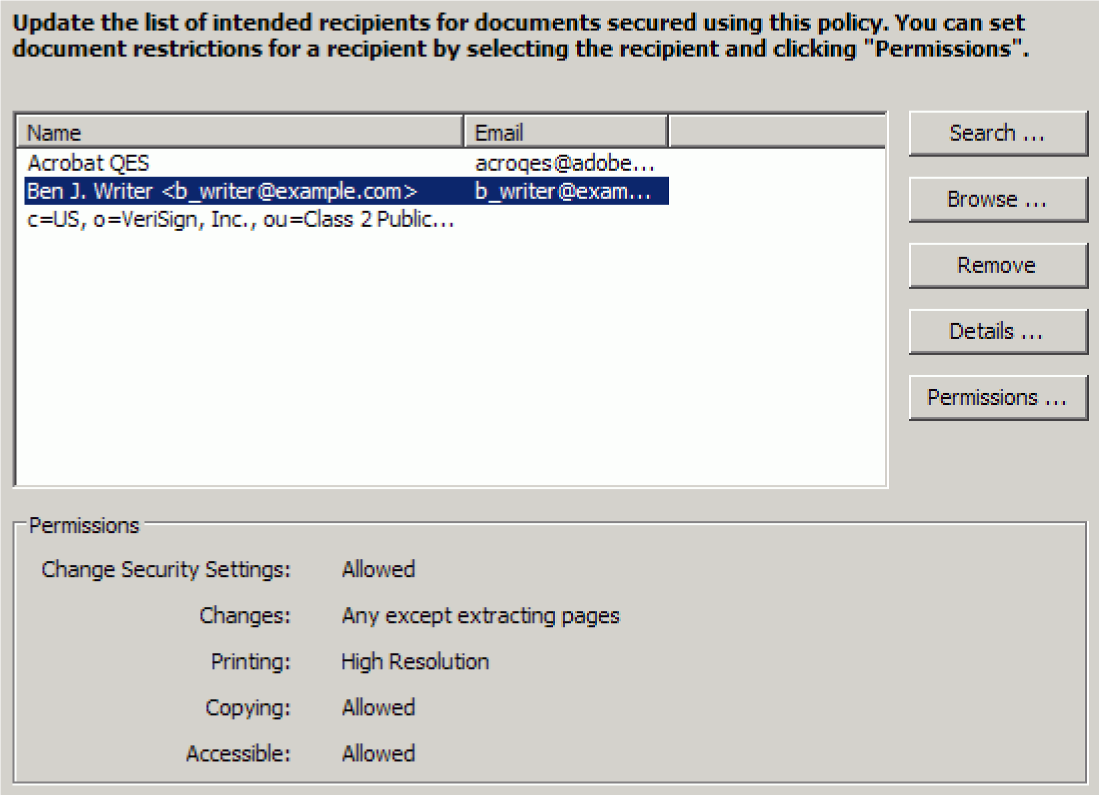
    Note: This option will not appear for users with only one digital ID.

12. Choose **OK**.

13. Add document recipients to the recipient list. You will be encrypting the document with each recipient's public key so that they can decrypt it. Choose from the following:
    - **Search** lets you search preconfigured directories for certificates on remote servers as well as in your local Trusted Identity list. Highlight one or more found digital IDs and choose **OK**.
    - **Browse** lets you search your computer for certificate files stored locally. Highlight one or more found digital IDs and choose **OK**.

    Tip: Admins typically preconfigure access to the company LDAP directory server so a certificate search can be automatically company-wide.

Figure 29 Adding recipients to a document with certificate security



Update the list of intended recipients for documents secured using this policy. You can set document restrictions for a recipient by selecting the recipient and clicking "Permissions".

Name	Email
Acrobat QES	acroqes@adobe...
Ben J. Writer <b_writer@example.com>	b_writer@exam...
c=US, o=VeriSign, Inc., ou=Class 2 Public...	

Permissions

Change Security Settings: Allowed

Changes: Any except extracting pages

Printing: High Resolution

Copying: Allowed

Accessible: Allowed

14. If you want to specify document permissions, do the following. Otherwise, skip to Step 16.

    1. Highlight one or more recipients. Different permissions can be set for different recipients. Select multiple recipients from the list by using the **Control** or **Shift** keys.

    2. Choose **Permissions**.

    3. When an alert appears stating that non-Adobe products may not respect these settings, choose **OK**.

    4. Check **Restrict printing and editing of the document and security settings**.

       Caution: Adobe products enforce permissions restrictions. However, not all third-party products fully support and respect these permissions. Encryption and therefore document access would likely not be impaired, but Adobe cannot guarantee that individual permissions settings will remain function. Recipients using such third-party products might be able to bypass some of your restrictions.

    Set the permissions as needed:

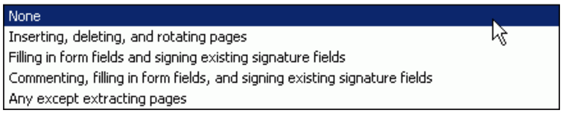
    1. **Printing Allowed**:

       - **None**: Prohibits printing.
       - **Low Resolution**: Limits printing to 150-dpi resolution. Printing may be slower because each page is printed as a bitmapped image. This option is only available if a high encryption level (Acrobat 5 or Acrobat 6) is selected. Low resolution also inhibits users from

printing a high quality document and using optical character recognition software to create a similar document with no security.

- **High Resolution**: Allows printing at any resolution. For example, you can direct high-quality vector output to PostScript and other printers that support advanced high-quality printing features.

2. **Changes Allowed**: Limits page-level editing, commenting, and form field interaction.



None
Inserting, deleting, and rotating pages
Filling in form fields and signing existing signature fields
Commenting, filling in form fields, and signing existing signature fields
Any except extracting pages

- **None**: Prevents users from changing the document, including filling in signature and form fields.
- **Inserting, deleting, and rotating pages**: Lets users insert, delete, rotate pages, and create bookmarks and thumbnail pages. This option is only available if a high encryption level is selected.
- **Filling in form fields and signing existing signature fields**: Lets users fill in forms and add digital signatures. This option doesn't allow users to add comments or create form fields.
- **Commenting, filling in form fields, and signing existing signature fields**: Lets users fill in forms and add digital signatures and comments.
- **Any except extracting pages**: Lets users change the document using any method listed in the **Changes Allowed** menu, except remove pages.

3. **Enable copying of text, images, and other content**: Allows file contents (excluding comments) to be copied. It also makes the content available to assistive technology devices such as screen readers. It also lets utilities that need access to the contents of a PDF file, such as Acrobat Catalog, get to those contents. This option is only available if a high encryption level is selected.

4. **Enable text access for screen reader devices for the visually impaired**: Only available if the option above is NOT checked. Lets visually impaired users read the document with screen readers. This option doesn't allow users to copy or extract the document's contents. This option is only available if a high encryption level is selected.

15. Choose **OK**.

16. Choose **Next**.

17. Choose **Finish**.

4.3.2 Creating Certificate Security for the Current Document

This workflow allows you to save the settings as a policy or discard them after they are applied. Use this method if you:

- Have access to the document recipient’s digital IDs.
- Do not need to save the settings as a policy.

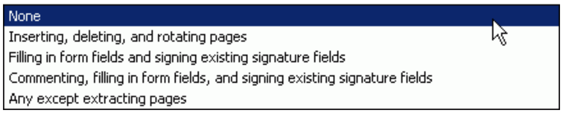
To apply certificate security to the current document:

1. Choose **File > Properties > Security tab**.
2. Select **Certificate Security** from the **Security Method** drop-down list.
3. Choose one of the following:
   - **Save these settings as a policy**: Choosing to save the settings as a policy activates the **Policy name** and **Description** fields. Once the wizard is completed, the settings are saved as a policy and added to the policy list in the **Advanced > Security** menu and the Policy Manager. If you are creating a policy, enter a policy name and optional description.
   - **Discard these settings after applying**: Choosing to discard the settings deactivates the **Policy name** and **Description** fields and no settings are saved.
4. Configure the **Select Document Components to Encrypt** panel:

   Note: Adobe products enforce permissions restrictions. However, not all third-party products fully support and respect these permissions. Encryption and therefore document access would likely not be impaired, but Adobe cannot guarantee that individual permissions settings will remain function. Recipients using such third-party products might be able to bypass some of your restrictions.

   Set the permissions as needed:

   1. **Printing Allowed**:
      - **None**: Prohibits printing.
      - **Low Resolution**: Limits printing to 150-dpi resolution. Printing may be slower because each page is printed as a bitmapped image. This option is only available if a high encryption level (Acrobat 5 or Acrobat 6) is selected. Low resolution also inhibits users from printing a high quality document and using optical character recognition software to create a similar document with no security.
      - **High Resolution**: Allows printing at any resolution. For example, you can direct high-quality vector output to PostScript and other printers that support advanced high-quality printing features.
   2. **Changes Allowed**: Limits page-level editing, commenting, and form field interaction.

      
      None
      Inserting, deleting, and rotating pages
      Filling in form fields and signing existing signature fields
      Commenting, filling in form fields, and signing existing signature fields
      Any except extracting pages

      - **None**: Prevents users from changing the document, including filling in signature and form fields.
      - **Inserting, deleting, and rotating pages**: Lets users insert, delete, rotate pages, and create bookmarks and thumbnail pages. This option is only available if a high encryption level is selected.
      - **Filling in form fields and signing existing signature fields**: Lets users fill in forms and add digital signatures. This option doesn't allow users to add comments or create form fields.

- **Commenting, filling in form fields, and signing existing signature fields**: Lets users fill in forms and add digital signatures and comments.
- **Any except extracting pages**: Lets users change the document using any method listed in the **Changes Allowed** menu, except remove pages.

3. **Enable copying of text, images, and other content**: Allows file contents (excluding comments) to be copied. It also makes the content available to assistive technology devices such as screen readers. It also lets utilities that need access to the contents of a PDF file, such as Acrobat Catalog, get to those contents. This option is only available if a high encryption level is selected.

4. **Enable text access for screen reader devices for the visually impaired**: Only available if the option above is NOT checked. Lets visually impaired users read the document with screen readers. This option doesn't allow users to copy or extract the document's contents. This option is only available if a high encryption level is selected.

5. Choose the encryption algorithm:
   - **128-bit RC4**: Compatible with Acrobat 6.0 and later as well as other non-Adobe and Adobe PDF clients such as Ghostscript and Apple Preview that have not implemented AES. RC4 has a smaller file size by about 32 bytes per stream.
   - **128-bit AES:** Compatible with Acrobat 7.0 and later. It is mandated for some U.S. government documents because it is more secure than RC4. AES has a bigger file size and adds up to 32 bytes per stream.
   - **256-bit AES**: Compatible with Acrobat 9.0 and later. Provides the highest level of encryption.

   Note: The RC4 encryption algorithm is unavailable if your administrator has configured your application to operate in FIPS mode.

6. Choose **Next**.

7. *The Digital ID Selection dialog may not appear. If it does not appear, go to the "add document recipients to the recipient list" step*: The digital ID selection dialog only appears if you have no digital IDs suitable for encryption or more than one. If you only have one digital ID suitable for encryption, then this dialog does not appear (for example, one ID is set as the default for encryption in the Security Settings Console). If the dialog does appear, select your digital ID that you will use to access this document in the future.

   Tip: While it is possible to apply certificate security without selecting your digital ID, doing so leaves you off of the recipient list and permanently locks you out of the document.

   If the required digital ID does not appear in the list, choose **Add Digital ID** (Figure 28).

8. If you have more than one digital ID, choose the digital ID persistence level.
   - Ask me which digital ID to use next time
   - Use this digital ID until I close the application
   - Always use this digital ID

   Note: This option will not appear for users with only one digital ID.

9. Choose **OK**.

10. Add document recipients to the recipient list. You will be encrypting the document with each recipient’s public key so that they can decrypt it. Choose from the following:

    - **Search** lets you search preconfigured directories for certificates on remote servers as well as in your local Trusted Identity list. Highlight one or more found digital IDs and choose **OK**.
    - **Browse** lets you search your computer for certificate files stored locally. Highlight one or more found digital IDs and choose **OK** (Figure 29).

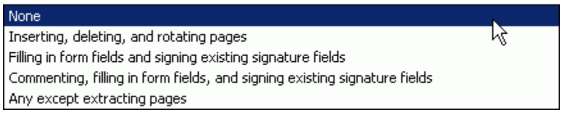
11. If you want to specify document permissions, do the following. Otherwise, skip to Step 13.

    1. Highlight one or more recipients. Different permissions can be set for different recipients. Select multiple recipients from the list by using the **Control** or **Shift** keys.
    2. Choose **Permissions.**
    3. When an alert appears stating that non-Adobe products may not respect these settings, choose **OK**.
    4. Check **Restrict printing and editing of the document and security settings**.

       Note: Adobe products enforce permissions restrictions. However, not all third-party products fully support and respect these permissions. Encryption and therefore document access would likely not be impaired, but Adobe cannot guarantee that individual permissions settings will remain function. Recipients using such third-party products might be able to bypass some of your restrictions.

    Set the permissions as needed:

    1. **Printing Allowed**:
        - **None**: Prohibits printing.
        - **Low Resolution**: Limits printing to 150-dpi resolution. Printing may be slower because each page is printed as a bitmapped image. This option is only available if a high encryption level (Acrobat 5 or Acrobat 6) is selected. Low resolution also inhibits users from printing a high quality document and using optical character recognition software to create a similar document with no security.
        - **High Resolution**: Allows printing at any resolution. For example, you can direct high-quality vector output to PostScript and other printers that support advanced high-quality printing features.
    2. **Changes Allowed**: Limits page-level editing, commenting, and form field interaction.


None
Inserting, deleting, and rotating pages
Filling in form fields and signing existing signature fields
Commenting, filling in form fields, and signing existing signature fields
Any except extracting pages

- **None**: Prevents users from changing the document, including filling in signature and form fields.
- **Inserting, deleting, and rotating pages**: Lets users insert, delete, rotate pages, and create bookmarks and thumbnail pages. This option is only available if a high encryption level is selected.

  - **Filling in form fields and signing existing signature fields**: Lets users fill in forms and add digital signatures. This option doesn't allow users to add comments or create form fields.
  - **Commenting, filling in form fields, and signing existing signature fields**: Lets users fill in forms and add digital signatures and comments.
  - **Any except extracting pages**: Lets users change the document using any method listed in the **Changes Allowed** menu, except remove pages.

 3. **Enable copying of text, images, and other content**: Allows file contents (excluding comments) to be copied. It also makes the content available to assistive technology devices such as screen readers. It also lets utilities that need access to the contents of a PDF file, such as Acrobat Catalog, get to those contents. This option is only available if a high encryption level is selected.

 4. **Enable text access for screen reader devices for the visually impaired**: Only available if the option above is NOT checked. Lets visually impaired users read the document with screen readers. This option doesn't allow users to copy or extract the document's contents. This option is only available if a high encryption level is selected.

12. Choose **OK**.

13. Choose **Next**.

14. Choose **Finish**.

4.3.3 Applying a Certificate Security Policy

If your certificate security settings already exist in a policy, apply those settings with Policy Manager:

1. Choose **Tools > Protection > Encrypt > Manage**.

2. Select a policy that uses certificate security.

3. Choose **Apply to Document**.

4. Save the document. New or changed settings do not appear in the user interface until the document is closed and reopened.

4.3.4 Applying a Certificate Security to a Group

Certificate security may be applied to more than one individual at a time. To apply security for multiple people, use the Search for Recipients dialog:

1. Configure certificate security as described in "Creating Certificate Security Settings" on page 38. When you are prompted to add document recipients to the recipient list, choose **Search**.

2. Select the Search Directories:

   - Check **Search all directories** to search all the directories you have configured in the Security Settings Console.
   - Uncheck **Search all directories** to search a specific directory. If you are just searching for individuals in your Trusted Identities list:

1. Choose *Trusted Identities* from the **Directories** drop-down list.

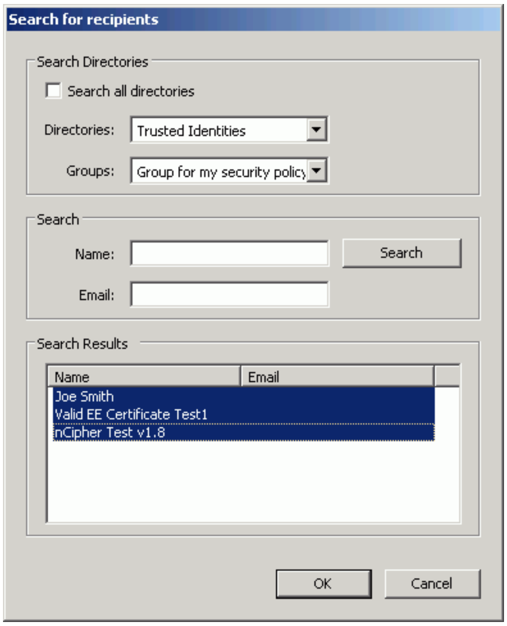
2. Select a group from the **Groups** drop-down list. All members of this group will appear in the **Search Results** field.s.

3. Enter a search name or email address.

4. Highlight one or more of displayed individuals.

5. Choose **OK** and continue configuring the security settings.

Figure 30 Searching for group contacts



Search for recipients

Search Directories

☐ Search all directories

Directories: Trusted Identities

Groups: Group for my security policy

Search

Name:

Email:

Search Results

Name	Email
Joe Smith	
Valid EE Certificate Test1	
nCipher Test v1.8	

4.3.5 Opening a Certificate-Protected Document

Password protected documents require that a user know the document open password to open it. If a permissions password has been set, that password can also be used to open the document.

To open a password protected document:

1. Open the document.

2. Enter the password associated with the digital ID used to encrypt the document.

3. Choose **OK**.

5 LiveCycle Rights Management Server Security

Adobe LiveCycle Rights Management Server (ALCRMS) security is only available to users with access to an Adobe LiveCycle Rights Management Server.

Tip: This document provides a cursory overview of the ALCRMS features. For information on configuring your application to use an Adobe LiveCycle Rights Management Server, log in to the server and use the help system.

ALCRMS's server-based security system provides a Web-based user interface for dynamic document control through the use of policies stored on a server. The policies enable centralized document management and event auditing. The documents that use those policies can reside anywhere. ALCRMS policies not only enable reusing security settings, but they also have let the author expire and revoke documents irrespective of how many copies were created or distributed. You can also maintain accountability and audit who opens protected documents.

ALCRMS can be configured to run with LDAP, ADS, and other enterprise systems so that user lists can be leveraged from an organization's existing information.

Using server-based security policies includes the following steps:

1. **Application configuration**: A system administrator configures the machine or the end user can do it manually. Server settings can also be sent via an .acrobatsecurity or an FDF file thereby enabling the end user to automatically import the requisite settings from a secure file. The administrator manages accounts, sets up organizational policies, and maintains the server.
2. **Policy configuration**: Reviewing the list of preconfigured organizational policies or creating a new user policy.
3. **Apply the policy and publish the document**: You apply a policy to the document with Acrobat's Policy Manager which can be accessed via the security main menu or through the Document Properties dialog. The policy server generates a license and an encryption key. Acrobat embeds the license in the document and encrypts it using the encryption key. You distribute the document or tell others where to find it.
4. **Viewing a document that has a policy applied**: When users try to open the document, they must authenticate their identities. The document is decrypted and opens with whatever permissions are specified in the policy.
5. **Auditing events and modifying access**: You audit document and usage history by logging in to the ALCRMS. You can modify access rights and user policies.

5.1 Configuring Servers

In most cases if you have access to an ALRM server then you're administrator will set up the server for you. There are three ways to set up an ALRM server:

- "Security Settings Import" on page 49

- "FDF File Import" on page 49
- "Configuring ALCRMS Settings Manually" on page 49

5.1.1 Importing ALCRMS Settings

5.1.1.1 Security Settings Import

An administrator may export the requisite security settings and provide you with a file to import. In this case, you will import any settings in the file according to the administrators instructions.

5.1.1.2 FDF File Import

Adobe LiveCycle Rights Management Server settings can be distributed via FDF files. Both users and administrators can import and export server settings in the same way as timestamp and directory server information is imported and exported.

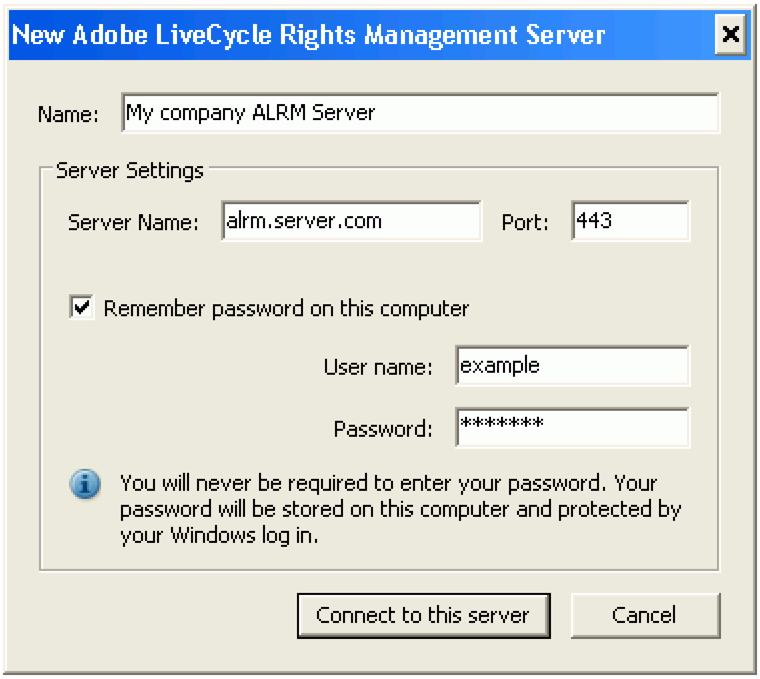
5.1.2 Configuring ALCRMS Settings Manually

Your server administrators will provide you with server connection details. Once these details are obtained, configure Acrobat to use the server.

To connect to a Adobe LiveCycle Rights Management Server:

1. Choose **Advanced > Security Settings**.
2. Select Adobe LiveCycle Rights Management Servers in the left-hand panel.
3. Choose **New**.
4. Enter the server settings:
   - **Name**: The server name.
   - **Server Name**: The server URL.
   - **Server Port**: The server port.
   - **Username**: The login username if required.
   - **Password**: The login password if required.
5. Choose **Connect to this Server**.
6. Choose **OK**.

Figure 31 ALCRMS Server Configuration



New Adobe LiveCycle Rights Management Server

Name: My company ALRM Server

Server Settings

Server Name: alrm.server.com Port: 443

☑ Remember password on this computer

User name: example

Password: ********

You will never be required to enter your password. Your password will be stored on this computer and protected by your Windows log in.

5.1.3 Managing your ALCRMS Account

To manage your ALCRMS Account:

1. Choose **Tools** > **Protection** > **More Protection** > **Rights Management** > **Manage My Account**.
2. If prompted, enter a username and password and choose **OK**.
3. Manage your account as described in the Adobe LiveCycle Rights Management Help documentation.

5.2 Working with Documents and ALCRMS Policies

5.2.1 Creating an ALCRMS Security Policy

ALCRMS policies are created using the server’s web interface. However, it is possible to launch that interface directly from Acrobat. Once the policy is created, return to Acrobat and choose **Finish** to add the policy to the policy list.

To create an ALCRMS user security policy:

1. Choose **Tools** > **Protection** > **Encrypt** > **Manage**.
2. Choose **New**.

3. Select **Use Adobe LiveCycle Rights Management**.

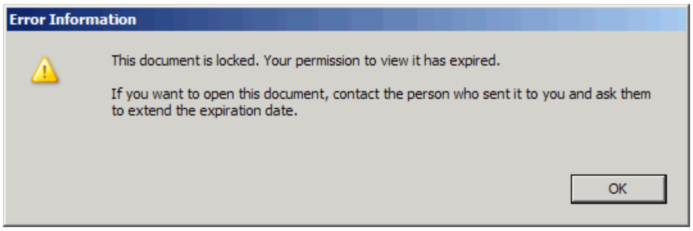
4. Log in to the server.

5. Navigate to the Policies page.

6. Enter a policy name and optional description.

7. Configure the **Validity period** panel. A document's validity period determines how long it will be accessible. When a recipient opens a document with an expired validity period, an alert appears stating that the document is locked (Figure 32).

Figure 32 Validity period expired alert

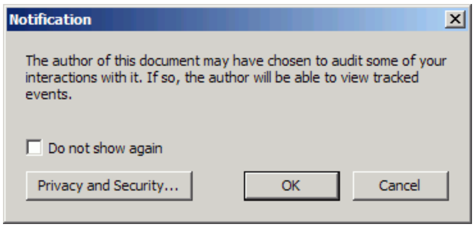


This document is locked. Your permission to view it has expired.

If you want to open this document, contact the person who sent it to you and ask them to extend the expiration date.

8. Choose **Yes** or **No** to **Audit Documents**. Auditing tracks events such as printing, modifying, viewing, closing, form filling, and signing documents.

Figure 33 Audit alert for ALCRMS security



The author of this document may have chosen to audit some of your interactions with it. If so, the author will be able to view tracked events.

☐ Do not show again

9. Set the **Auto-Offline lease period** to specify how long the document can be viewed offline before a user must synchronize with Adobe LiveCycle Rights Management Server.

10. Choose **Save**.

11. Exit the Web console and return to Acrobat.

12. Choose **Finish**.

5.2.2 Applying ALCRMS Security

Your ALCRMS policies will appear in Acrobat policy list.

To apply an ALCRMS policy:

1. Choose **Tools** > **Protection** > **Encrypt** > **Manage** .

2. Highlight a policy.

3. Choose **Apply to Document**.

5.2.3 Synchronizing a Document for Offline Use

Synchronizing a document for offline use allows you to get the latest version so that you can access it when you are not connected to the network. To synchronize a document:

1. Choose **Tools** > **Protection** > **More Protection** > **Rights Management** > **Synchronize for Offline**.
2. If prompted, enter a username and password and choose **OK**.

5.2.4 Revoking a Document

To revoke a document so that it cannot be viewed by anyone:

1. Choose **Tools** > **Protection** > **More Protection** > **Rights Management** > **Revoke**.
2. If prompted, enter a username and password and choose **OK**.
3. Enter the revocation details as described in the Adobe LiveCycle Rights Management Help documentation.
4. Choose **OK**.

5.2.5 View a Document’s Audit History

To view a document’s audit history:

1. Open the document you would like to track.
2. Choose **Tools** > **Protection** > **More Protection** > **Rights Management** > **View Audit History**.
3. If prompted, enter a username and password and choose **OK**.
4. View the audited events as described in the Adobe LiveCycle Rights Management Help documentation.

6 Registry and plist Settings

This chapter describes the application preferences for the Acrobat family of products' digital signature and document security features. Before continuing, you should know that:

- The tables and examples use the Windows registry. Most are applicable to Macintosh, Unix, and Linux systems.
- The root security directory for registry settings stored on a per-user basis are at: **`HKEY_CURRENT_USER\Software\Adobe\<application>\<version number>\Security\`**.
- The examples use Acrobat; other applications may provide different menu options.
- The security preferences folder does not appear in the registry until after the Acrobat product installation and a security feature is used. Subdirectories also appear as the code is exercised.

Caution: Adobe strongly recommends that you do not make changes to the registry unless you are knowledgeable about editing and troubleshooting Microsoft Windows registry settings. Improper use of this feature can result in the corruption of critical system files.

6.1 Setting Basic Client and Workflow Preferences

6.1.1 Preventing End-User Modification

While many lockable preferences have a matching corollary stored in HKCU, their editability via the user interface is controlled by their boolean lockdown counterpart in HKLM. When marked as uneditable, the application user interface item associated with that preference is disabled. The hierarchy within **`FeatureLockdown`** is typically the same as the one under the HKCU **`Security`** directory.

The Adobe Customization Wizard provides a UI for modifying some of these keys when tuning the client installer for Windows prior to deployment. However, because some lockable keys are not exposed in the wizard, it is often simpler to modify the keys manually and then use the Wizard to drag and drop the configured registry directories to the installer.

To lock down features:

1. Navigate to:
   - **7.x**: HKEY_LOCAL_MACHINE\SOFTWARE\Adobe\<product>\<version>\FeatureLockDown
   - **8.x and later**: HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Adobe\<product>\<version>\FeatureLockDown
   - **64 bit Windows**: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Adobe\FeatureLockdown.
2. Create a directory path that mirrors the path in HKCU. For example, when configuring a digital signature key, create a `cSecurity` directory.

Tip: The HKLM path usually mirrors the HKCU path, but is sometimes different. For details about all keys besides those listed here, refer to the Preference Reference for the Acrobat family of Products.

3. Add the requisite subdirectories. Use the same path used for the settings described in the rest of this chapter. For example, create a directory `cPPKLite` to set `bAllowPasswordSaving"=dword:00000000`. See Example 6.5.

4. **Predeployment installer tuning via the Wizard**: When you finish your other registry modifications, use the wizard UI to drag and drop the FeatureLockDown directory from My Computer to the Destination Computer.

Tip: Most of the lock down keys are booleans indicating whether the preference is editable by end users. For more detail about a particular setting, refer to the Preference Reference for the Acrobat family of Products. When looking for a key, search for the key name without the data type prefix. For example, search for `PrivKey` and NOT `bPrivKey` since that setting is not a boolean in HKCU.

Table 8 Registry preferences: subject to feature lockdown

Preference	Feature	Description
bAllowPasswordSaving	Various	Caches passwords so they don't have to be re-entered when accessing digital IDs, policies, and other features that use passwords. HKLM and HKCU
bPrivKey	Certificate handling	Prevents a user from changing the security handler used for signing and certificate security. HKLM and HKCU
bVerify	Signature validation	Prevents a user from changing the security handler used for the default signature verification method. HKLM and HKCU
bVerifyUseAlways	Signature validation	Qualifies the use of aVerify. HKLM and HKCU
bValidateOnOpen	Signature validation	Forces signature validation when a document opens. HKLM and HKCU
bReqRevCheck	Signature validation	Requires revocation checking to behave as specified. HKLM and HKCU
bReasons	Signing	Prevents users from modifying the reasons setting. v8.1: If locked and cReasons if empty, bAllowSigningReasons is 0 and read only. If locked and cReasons has values, then bAllowSigningReasons is true and read only. HKLM only. HKLM\Software\Policies\Adobe\<product>\<version>\FeatureLockdown\cSecurity\cPubSec
bSuppressStatusDialog	Signing	Deprecated since 8.0. Prevents the Document Status dialog from appearing when a certified document opens. HKLM and HKCU

Table 8 Registry preferences: subject to feature lockdown

Preference	Feature	Description
bWinCacheSession Handles	Signing	(v 8.1 Windows only) Default: 1 **Path**: cPPKHandler Specifies whether to retain CSP handles when a user authenticates to a digital ID. If true, a user does not have to reauthenticate to use the ID unless they log out or the session ends. The impact of this preference will vary based on the CSP in use; however, the setting does not affect the Windows CSPs. HKLM only
bAllowInvisibleSig	Signing and document security	Prevents user from signing with an invisible certification signature. Disables the menu option in the signing menus. HKLM and HKCU
bAllowAPSConfig	Document security	v 8.1 (Windows only) Default: 1 Prevents a LiveCycle Right Management Server from being configured by disabling the menu option in the Security Settings Console. HKLM only

Example 6.5: Lockdown keys

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Adobe\Adobe Acrobat\8.0\FeatureLockDown\
cDefaultLaunchAttachmentPerms]"tBuiltInPermList"="version:1|.ade:3|.adp:3|.app:3|.asp:3|.bas:3|.bat:
3|.bz:3|.bz2:3|.chm:3|.class:3|.cmd:3|.com:3|.command:3|.cpl:3|.crt:3|.csh:3|.desktop:3|.exe:3|.fxp:3|.gz:3|.
hex:3|.hlp:3|.hqx:3|.hta:3|.inf:3|.ini:3|.ins:3|.isp:3|.its:3|.job:3|.js:3|.jse:3|.ksh:3|.lnk:3|.lzh:3|.mad:3|.maf:3|.
mag:3|.mam:3|.maq:3|.mar:3|.mas:3|.mat:3|.mau:3|.mav:3|.maw:3|.mda:3|.mde:3|.mdt:3|.mdw:3|.mdz:3|.
msc:3|.msi:3|.msp:3|.mst:3|.ocx:3|.ops:3|.pcd:3|.pi:3|.pif:3|.prf:3|.prg:3|.pst:3|.rar:3|.reg:3|.scf:3|.scr:3|.sct:3|.
sea:3|.shb:3|.shs:3|.sit:3|.tar:3|.tgz:3|.tmp:3|.url:3|.vb:3|.vbe:3|.vbs:3|.vsmacros:3|.vss:3|.vst:3|.vsw:3|.
webloc:3|.ws:3|.wsc:3|.wsf:3|.wsh:3|.zip:3|.zlo:3|.zoo:3|.pdf:2|.fdf:2"

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Adobe\Adobe Acrobat\8.0\FeatureLockDown\
cDefaultLaunchURLPerms]"tSchemePerms"="version:1|shell:3|hcp:3|ms-help:3|ms-its:3|ms-itss:3|its:
3|mk:3|mhtml:3|help:3|disk:3|afp:3|disks:3|telnet:3|ssh:3|javascript:1|vbscript:1|acrobat:2|mailto:2|file:2"

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Adobe\Adobe Acrobat\8.0\FeatureLockDown\
cDocumentStatus]"bSuppressMessageBar"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Adobe\Adobe Acrobat\8.0\FeatureLockDown\cSecurity\
cDigSig]"bValidateOnOpen"=dword:00000000

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Adobe\Adobe Acrobat\8.0\FeatureLockDown\cSecurity\
cEDC]"bAllowAPSConfig"=dword:00000000

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Adobe\Adobe Acrobat\8.0\FeatureLockDown\cSecurity\
cHandlers]
"bVerify"=dword:00000000
"bPrivKey"=dword:00000000
"bVerifyUseAlways"=dword:00000000

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Adobe\Adobe Acrobat\8.0\FeatureLockDown\cSecurity\
cPubSec]"bReasons"=dword:00000000
```

6.1.2 Importing Security Settings

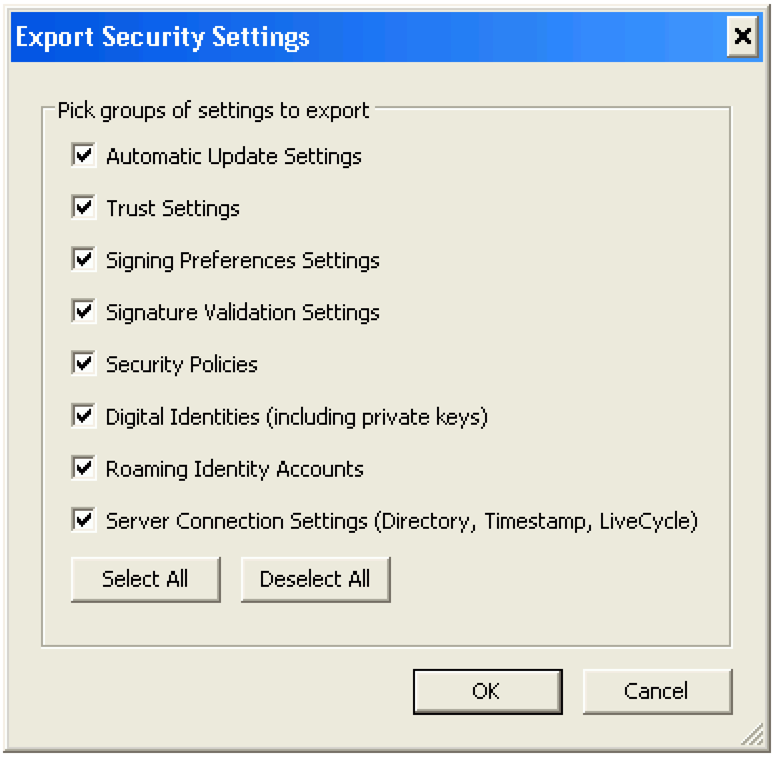
One of Acrobat 9.0's major new security features includes the ability to import and export security settings via an .acrobatsecuritysettings files, thereby enabling easier version upgrades as well as configuration of multiple machines.

The security settings import/export features offers several advantages over FDF files:

- Most document security and digital signature related settings can be encapsulated in an `acrobatsecuritysettings` file whereas FDF could only transport one setting type and a time and could not encapsulate registry settings at all.
- One file can be used instead of many files.
- Trust can be assigned to imported on the fly, thereby simplifying workflows.
- Files can be signed and encrypted.

Use security settings files to backup and restore settings, to distribute settings in a workgroup or enterprise, and to send specific information to another user. Importing settings simply involves importing a file from a network (including automatically from a server) that has been exported from Acrobat and has then been made available from a trusted source.

Figure 23 Security settings: Export dialog



Export Security Settings

Pick groups of settings to export

☑ Automatic Update Settings

☑ Trust Settings

☑ Signing Preferences Settings

☑ Signature Validation Settings

☑ Security Policies

☑ Digital Identities (including private keys)

☑ Roaming Identity Accounts

☑ Server Connection Settings (Directory, Timestamp, LiveCycle)

The following options are available:

- Specifying whether or not to poll a server for settings to import at regular intervals.
- Configuring whether or not the user should grant permission prior to installing new settings.
- Specifying a particular certificate so the signed settings will only be imported from a trusted source.

Table 9 Registry preferences: Security setting import

Name	Type	Description
bAskBeforeInstalling	bool	(v. 9.0) Default: 1 **Path**: <security root>\DigSig\ Maps to GUI item: **Ask before installing** checkbox in Preferences > Security > Security Settings panel. Specifies whether the settings should be imported silently or Acrobat should ask permission from the user.
bLoadSettingsFromURL	bool	(v. 9.0) Default: 0; 10: 1 **Path**: <security root>\DigSig\ Maps to GUI item: **Load security settings from a server** in Preferences > Security > Security Settings panel. Specifies whether or not security settings should be automatically imported from a server at the specified time interval.
iCheckEvery	int	(v. 9.0) Default: 2419200 **Path**: <security root>\DigSig\ Maps to GUI item: **Check every** radio buttons in Preferences > Security > Security Settings panel. The polling interval to check the server for updated settings to import. Specifies the number of seconds it should wait between checks for updates. The default value is 30 days. The options are: **604800**: 1 week **1209600**: 2 weeks **2419200**: 1 month **7257600**: 3 months
iResourceID	int	(v. 9.0) Default: null **Path**: <security root>\DigSig\ An internally used number created by Acrobat when it first sets up the “resource” pointed to by the URL. It is not user customizable.
tLoadSettingsURL	text	(v. 9.0) Default: null **Path**: <security root>\DigSig\ Maps to GUI item: **URL** text box in Preferences > Security > Security Settings panel. The server URL where the acrobatsecuritysettings file to import resides.
tLoadSettingsCERT	cab	(v. 9.0) Default: null (allows any certificate) **Path**: <security root>\DigSig\ Maps to GUI item: **Settings must be signed by** field in Preferences > Security > Security Settings panel. Specifies a certificate that must be used to sign the imported security settings file. The value is a hexadecimal string corresponding to the SHA-1 hash of the certificate used to sign the settings file.
xdata	string	(v. 9.0) Default: null **Path**: <security root>\DigSig\cLastChecked Binary data used for internal purposes. It is not set during installation or for tuning pre-deployment clients. It can safely be deleted in an existing environment.

6.1.3 Digital ID Management

Digital ID preferences control or record the user's personal digital ID behavior for certificate security and/or digital signatures. Some preferences are not customizable and are listed only for information purposes (Table 10). You can set the following:

- "Allowing Self Sign Digital IDs" on page 49
- "PKCS#11 Configuration" on page 50
- "Roaming ID Configuration" on page 50

6.1.3.1 Specifying extended certificate information

Table 10 Registry preferences: Digital ID

Name	Type	Description
cExtendedCertInfo	cab	(v 8.0) Default: null **Path**: <security root>\cPubSec Contains a subkey for each certificate with extended information provided by attribute certificates The subkeys take the form c{DIGEST} where {DIGEST} is a SHA-1 digest of the associated certificate's public key encoded as hexadecimal. For example, \cPubSec\ cExtendedCertInfo\cAD6716326BDAC87628DFAD6716326. Each subkey contains the friendly name, related ID card, and associated attribute certificates.
cCertIssuerInfo	cab	(v 8.0) Default: null **Path**: <security root>\cPubSec Contains a subkey for each certificate with extended information. The subkeys take the form c|{DN} where {DN} is the issuer certificate's distinguished name. For example, \ cPubSec\cCertIssuers\c|cn=Adobe Systems, o=Acrobat Engineering. Each subkey contains the associated ID card for this issuer certificate.

6.1.3.2 Allowing Self Sign Digital IDs

By default, users can create self signed digital IDs. However, if you would like to prevent users from creating their own IDs, turn this feature off. Disabling this option prevents users from selecting **Create a self-signed ID** option in **Add ID** workflows.

Table 11 Registry preferences: Self-signed digital ID

Name	Type	Description
bSelfSignCertGen	bool	(v 7.0) Default: 1 **Path**: <security root>\cPubSec Maps to GUI item: **Create a self-signed digital ID for use with Acrobat**. Turns on and off the **Create a self-signed ID** option in **Add ID** workflows so that a user can create a self signed digital ID.

6.1.3.3 PKCS#11 Configuration

The key `cAdobe_P11CredentialProvider\` contains a list of P11 modules the user has loaded by choosing **Attach Modules** in the Security Settings console. By specifying a valid path to a PKCS#11 DLL, modules can be pre-attached to installed clients. Because various errors appear as a result of a bad filename or pointing to a dll that is not a valid PKCS#11 module, test the settings and file before distributing them.

The following options are available:

- Preconfiguring the key when tuning the installer and distributing the module file or when modules are already installed.
- Setting the default browse path in which to look for additional modules.

> **Note:** For Reader X (10.0), not all PKCS#11 devices may work with Protected Mode (PM) enabled. However, in most cases, they do. Installation of such devices usually involves disabling Protected Mode, installing the driver, restarting the application, and then re-enabling Protected Mode. For the latest information about PM compatibility with certain features, see http://kb2.adobe.com/cps/860/cpsid_86063.html.

Table 12 Registry preferences: PKCS#11

Name	Type	Description
cModules	Array of strings	(v. 7.0) Default: null **Path**: <security root>\cASPKI\cAdobe_P11CredentialProvider\ Maps to GUI item: **Attach Module** in Security Settings console. Array of dynamic library paths to PKCS#11 modules. These may not necessarily be full paths but just something that the OS dynamic library loading functions will accept. For example, t0 may be a path to `C:\WINDOWS\system32\dkck201.dll`.
cP11Path	ASPath	(v 7.0) Default: null **Path**: <security root>\cPubSec Stores the last folder in which the user browsed for a P11 module. The next time the user goes to add a P11 module browsing starts in that folder.
cP11Credentials	cab	(v 7.0) Default: null **Path**: <security root>\cPPKHandler Contains an array of subcabs for all known PKCS#11 digital IDs. The format is as follows: **xCert**: Binary value of the certificate **xTokenKey**: Binary value generated from the IDs PKCS#11 token. The binary value is generated with the following method: Initialize SHA-1 digest, add the digest the value of the token label, token manufacturer, token model, and token serial. Finish the SHA-1 digest operation. The resulting 20-byte value is the token key.

6.1.3.4 Roaming ID Configuration

These preferences allow you to configure an application to use roaming IDs. While the needed configuration can be handled through the user interface by end users, you can set the following:

- **Specifying a Default Roaming ID Server**: When a user adds a roaming ID account through the GUI, a dialog asks for a friendly name and a server URL. If no other accounts have been configured and `cDefaultServerInfo` exists in the preferences (Table 13), its values populate both the friendly server name and URL fields in the Add a Roaming ID dialog.

- **Specify one or more authentication methods**: See "Roaming ID Authentication Mechanisms" on page 67.

Table 13 Registry preferences: Roaming ID server

Name	Type	Description
tServerName	text	(v. 8.0) Default: null **Path**: <security root>\cASPKI\cAdobe_RoamingID\cDefaultServerInfo A user friendly roaming ID server name.
tURL	text	(v. 8.0) Default: null **Path**: <security root>\cASPKI\cAdobe_RoamingID\cDefaultServerInfo A roaming ID server URL.

Roaming ID Provider Persistent Storage

These preferences store roaming ID server data. Some values are provided by the user and some are provided by the server.

Note: These keys do not need to be customized and are provided for informational purposes only.

Table 14 Registry preferences: Roaming ID account

Name	Type	Description
cAccounts	cab	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version> Contains entries for user accounts on roaming ID servers that the provider knows about. Every account is identified by a unique 9-character key such as cAB2CFECD.
cRecentServerURLs	cab	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version> Contains an array of roaming ID server URLs recently entered by the user.

Roaming ID Server-Set Preferences

The preferences in Table 15 are typically created as a result of communications with a roaming ID server.

Note: Whether or not you customize these settings is determined by the needs or your particular implementation.

Table 15 Registry preferences: Roaming ID user account (set by the server)

Name	Type	Description
tFriendlyName	text	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version> The friendly name created by the user for user interface display purposes.

Table 15 Registry preferences: Roaming ID user account (set by the server)

Name	Type	Description
tServerURL	text	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version>\accounts\<accountname> The roaming ID server URL on which this account exists.
cCredentials	cab	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version> The value is provided by the server. An array of certificates corresponding to digital IDs available through this account. The certificates are in the binary X.509 format.
cSAML_Assertion	cab	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version> The value is provided by the server. Holds an encrypted SAML assertion obtained during last successful authentication. Possession of this assertion is proof of a user's identity. Therefore, the assertion is encrypted using 256-bit AES algorithm in CBC mode. The encryption key is stored in Microsafe database that is protected by the OS login. There are two binary entries under the cSAML_Assertion cab: xEncryptedData contains the encrypted assertion, 'xIV' contains the initialization vector used by the AES encryption algorithm for this assertion.
tSAML_Assertion_ Expiration	text	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version> The value is provided by the server. Holds the time after which roaming ID provider will not attempt to use the SAML assertion stored in cSAML_Assertion. This time is calculated when assertion is first obtained and takes into account clock difference between the client machine and the server that generated the assertion. Time is represented in BER GeneralizedTime format without the type and length octets.
tSAML_Assertion_ Source	text	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version> The value is provided by the server. Holds the URL of the authentication server from which the SAML assertion stored in cSAML_ Assertion was obtained.
tSAML_Name_ Value	text	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version> The value is provided by the server. SAML_NAME_<Value, Format, Qualifier> comprise the subject name identifier taken from the SAML assertion received during the account's last user authentication. The identifier is essentially a machine-readable user name that is unaffected by the choice of authentication mechanisms.
tSAML_Name_ Format	text	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version> The value is provided by the server. SAML_NAME_<Value, Format, Qualifier> comprise the subject name identifier taken from the SAML assertion received during the account's last user authentication. The identifier is essentially a machine-readable user name that is unaffected by the choice of authentication mechanisms.

Table 15 Registry preferences: Roaming ID user account (set by the server)

Name	Type	Description
tSAML_Name_ Qualifier	text	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version> The value is provided by the server. SAML_NAME_<Value, Format, Qualifier> comprise the subject name identifier taken from the SAML assertion received during the account's last user authentication. The identifier is essentially a machine-readable user name that is unaffected by the choice of authentication mechanisms.
tSASL_Mechanism	text	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version> The value is provided by the server. The SASL id of the authentication mechanism. Some authentication implementations may store user data. For example, a user name and password mechanism may store the username so that only the password needs to be entered during consequent authentications.
tSASL_UserName	text	(v 8.0) Default: null **Path**: <security root>\cPPKHandler\cRC<version> The value is provided by the server. The mechanism-specific persistent data. Some authentication implementations may store user data. For example, a user name and password mechanism may store the username so that only the password needs to be entered during consequent authentications.

Kerberos Authentication Mechanism

This option is only relevant if the ASSP-Kerberos SPI is selected as described in "Roaming ID Authentication Mechanisms" on page 67.

Table 16 Registry preferences: Kerberos

Name	Type	Description
sServiceName	string	(v. 8.0) Default: ASSP **Path**: <security root>\cASPKI\cKerberos_AuthMechanism The administrator-specified roaming ID Kerberos service name. If the key is not present, the default value of `ASSP` is assumed. If the key is present and the value is empty string, Acrobat asks the roaming ID service for it's Kerberos service name. This method is not secure and enterprises are advised not to use this option.

6.1.3.5 Setting Digital ID Defaults

Most digital ID default values are set by the application when a user first uses an ID or manually specifies a default value in the Security Settings Console. Moreover, since user actions will overwrite some preconfigured value an administrator might provide, setting many of these properties is usually not worthwhile. However, it is possible and the following options are available:

- Specifying a default URL to obtain a new digital ID. This value is NOT overwritten by user actions.
- Listing a set of attribute certificates.
- Specifying a default signing ID. This value is end user-specific.
- Specifying a default encryption ID. This value is end user-specific.
- Customizing a default directory server used to locate certificates that can be imported into the Trusted Identity Manager.

Caution: Acrobat 9.0 users who configure a 3rd party security handler plugin may find that their non-default choice does not stick if the plugin calls `PSUNregisterHandler()`. That is, each time Acrobat restarts, the non-default security handler choice is lost. To fix the problem, change the plugin code to not call `PSUNregisterHandler()`.

Table 17 Registry preferences: cPPKHandler

Name	Type	Description
xDefEnrollmentURL	text	(v 7.0) Default: null **Path**: <security root>\cPubSec **Maps to GUI item**: **Enroll at an online CA** The destination URL when the user selects **Enroll at an online CA** while adding a new digital ID. unless a different URL is specified as seed-value on the signature field being signed).
cDigitalIDFiles	cab	(v 7.0) Default: null **Path**: <security root>\cPPKHandler Contains an array of subcabs for all application-known digital ID files. The format is as follows: **cPath**: The path of to the digital ID file. **cCredentials**: An array of certificates that have corresponding private keys in the file. **cCertificates**: An array of certificates that are in the file but do not have an associated private key (usually CA certs). Certificates are stored as binary data.
cACs	cab	(v 7.0) Default: null **Path**: <security root>\cPPKHandler Contains a set of attribute certificates as binary data. Each certificate is indexed with an integer 0 to N. The value is only set when a user imports attribute certificates.
tCredProvider	text	(v 7.0) Default: null **Path**: <security root>\cPPKHandler\cCredSign Identifies credential service provider interface for the default signing digital ID. The value is set when a user opens the Security Settings Console and specifies a default signing ID. The value depends on the type of selected ID. For example, setting a self signed digital ID would result in a value of Adobe_FileCredentialProvider. See also xCertSHA1
xCertSHA1	text	(v 7.0) Default: null **Path**: <security root>\cPPKHandler\cCredSign Identifies the default signing digital ID by its SHA1 hash of the public key. The value is set when a user opens the Security Settings Console and specifies a default signing ID. See also tCredProvider

Table 17 Registry preferences: cPPKHandler

Name	Type	Description
tCredProvider	text	(v 7.0) Default: **Path**: <security root>\cPPKHandler\cCredCrypt Identifies credential service provider interface for the ASPKI provider which exposes this digital ID. The value is set when a user opens the Security Settings Console and specifies a default signing ID. The value depends on the type of selected ID. For example, setting a self signed digital ID would result in a value of Adobe_FileCredentialProvider. See also xCertSHA1.
xCertSHA1	text	(v 7.0) Default: null **Path**: <security root>\cPPKHandler\cCredCrypt Identifies the default encryption digital ID by its SHA1 hash of the public key. The value is set when a user opens the Security Settings Console and specifies a encryption signing ID. See also tCredProvider
bCustomPrefsCreated	bool	(v 7.0) Default: false **Path**: <security root>\cPPKHandler Indicates whether a custom certificate specific preference (e.g. Identrus) has already been created and written to the registry. If true, it doesn’t get created again. Deleting or setting this key to 0 forces Acrobat to recreate custom certificate preferences after which it will reset this key to 1.
aDefDirectory	ASAtom	(v 7.0) Default: Adobe.PPKMS.ADSI.dir0 **Path**: <security root>\cPPKHandler Maps to GUI item: Setting a default search directory affects the UI in two places: A star appears next to the default directory in the Security Settings Console and the directory is moved to the top of the directories drop down list in the Trusted Identities Manager’s Search for Recipients dialog. Default directory to use when searching for digital IDs. On Windows, the Adobe.PPKMS security handler provides access through the Microsoft Active Directory Script Interface (ADSI) to all the directories the user created in the Security Settings Console. These directories are named in the format of <directory handler> + <index>. For example, Adobe.PPKMS.ADSI.dir0, Adobe.PPKMS.ADSI.dir1, and so on. Unsupported for Linux and Macintosh.

6.1.3.6 Digital ID Import and Export Paths

The digital ID default path preferences point to the application security folder. For example, `C:\Documents and Settings\<user name>\Application Data\Adobe\Acrobat\8.0\Security`. The path is used when the user imports or exports an ID from the Security Settings Console. Since the application remembers the last accessed directory, if a user chooses a different directory, that action will overwrite the preconfigured value an administrator might provide.

The following options are available:

- Specifying a default path for exporting and importing digital ID certificates (does not include private keys).
- Specifying a default path for saving newly created digital ID files.

Table 18 Registry preferences: Digital ID path

Name	Typ e	Description
cExportPath	path	(v 7.0) Default: The applications security folder. For example, C:\Documents and Settings\<user name>\Application Data\Adobe\Acrobat\8.0\Security **Path**: <security root>\cPubSec Default path for exporting credentials. Used by all security plugins.
cImportPath	path	(v 7.0) Default: The application security folder. For example, C:\Documents and Settings\<user name>\Application Data\Adobe\Acrobat\8.0\Security **Path**: <security root>\cPubSec Default path for importing credentials. Used by all security plugins.
cEmbedded FilePath	path	Deprecated (v 7.0) Default: The application security folder. For example, C:\Documents and Settings\<user name>\Application Data\Adobe\Acrobat\8.0\Security **Path**: <security root>\cPubSec Default path for embedded files. The path last chosen for extracting an embedded file from a WebBuy FDF. The first time an embedded file is extracted from an FDF the user is asked where to save it.
cProfilePath	path	(v 7.0) Default: The applications security folder. For example, C:\Documents and Settings\<user name>\Application Data\Adobe\Acrobat\8.0\Security **Path**: <security root>\cPubSec Default path for storing profile files such as PKCS#12 files. This is used both when creating new digital ID files and when browsing for existing files. Used by all security plugins.

6.1.4 Certificate Management

Signing and certificate security workflows require that users obtain and trust other people's certificates. They will use those certificates to trust someone else's signature and to encrypt documents for them. Administrators have several options for facilitating this process. For details, see:

- "Windows Integration" on page 56
- "Trusted Identity List Configuration" on page 57
- "Custom Certificate Preferences" on page 58

6.1.4.1 Windows Integration

While Acrobat has its own store, the Windows store may already contain needed certificates or your enterprise may simply be a Windows shop. Windows integration allows end users to search for and use certificates in the Windows Certificate Store.

End users can configure their application for Windows integration through the application's Preference panel. Configuration options allow users to search the Windows store from the Trusted Identity Manager (through the **Search** button), set trust levels for any found certificate, and choose which certificates to use for encryption (once the certificate is located and added to the Trusted Identity Manager). If a user has a personal ID in the Windows store, it appears in the Security Settings Console automatically without any special configuration.

Administrators can control whether clients can access MSCAPI through Acrobat so that users can find, use, and set trust levels for Windows certificates. The following options are available:

- Adding the Windows Certificate Store as a searchable repository with `bCertStoreImportEnable`.
- Setting separate trust levels for approval and certification signatures.
- Preventing end user modification of certificate trust levels.
- Tuning the service provider interface:
  - Certificate Providers (for Signing and Decryption)
  - Revocation Checker Providers
  - Signature Validation Directory Providers

Table 19 Registry preferences: MSCAPI provider

Name	Type	Description
iMSStoreTrusted	int	(v. 7.0) Default: 0 **Path**: <security root>\cASPKI\cMSCAPI_DirectoryProvider Maps to GUI item: **Validating Signatures** and **Validating Certified Documents**. Controls whether or not certificates in the Windows Certificate Store are trusted for signing and certifying. **00: None.** **60**: **Validating Signatures** **62**: **Validating Signatures** and **Validating Certified Documents**.
bCertStoreImportEnable	bool	(v 7.0) Default: 0 **Path**: <security root>\PPKHandler Maps to GUI item: **Enable searching the Windows Certificate Store for certificates other than yours** If true, then users can import from MSCAPI certificate stores into their Trusted Identity Manager.

6.1.4.2 Trusted Identity List Configuration

The trusted identity list contains all of a users imported certificates that they use for validating someone else's signature or encrypting a document for them. The list is maintained and managed via the Trusted Identity Manager; however, administrators can preconfigure applications to use non-default list files, add certificates from the Windows, store, and so on.

The following options are available:

- Creating a custom filename/file for the trusted identity list.
- Specifying a non-default security handler to control Trusted Identity Manager functions. For details, see `aAddressBook` in Table 23.
- Adding the Windows Certificate Store as a searchable repository with `bCertStoreImportEnable`. For details, see Table 19.
- Turning off and on the ability to automatically download certificates sent by Adobe to users over the internet via bLoadSettingsFromURL.

Table 20 Registry preferences: Trusted Identity Manager

Name	Type	Description
tAddressBook	text	(v 7.0) Default: addressbook.acrodata **Path**: <security root>\cPubSec The filename the Trusted Identity Manager uses to read and write addressbook data.
bLoadSettingsFromURL	bool	(v 9.0) Default: 1 **Path**: <security root>\cDigSig\cAdobeDownload Maps to GUI item: **Load trusted root certificates from an Adobe server** which may be viewed by choosing **Preferences > Trust Manager**. Provides for the automatic download and installation of certificates which are automatically configured as a trust anchor.

6.1.4.3 Custom Certificate Preferences

The **`cCustomCertPrefs`** directory provides the means to use certificate-specific settings to modify application behavior when it encounters a particular certificate. As the application builds a certificate chain, it compares the information it finds in the certificate with that in the registry to see if there is a match. If there is a match, the custom settings are used to override the application's default behavior. Certificates that chain up to a CA that match those configured here or that contain recognized extensions will use the preferences set in this directory.

You can use any of the preferences available in **`\cASPKI`** in your customized preference under **`\cASPKI\cASPKI\cCustomCertPrefs`**. Custom certificate preferences are specified differently than when they are used globally under `\ASPKI`. For example, The naming and path convention is always `c<key>:c<index>:<type>Value`, where the global preference would be `sSignCertOID` and the custom preference would be `cSignCertOID` (the data type is associated with the `Value` subkey rather than the key.

Certificates are identified by creating a hash of a unique identifier and appending it to **`c`** such as **`c312E322E3834302E3131343032312E310000`**. When the application finds in a certificate chain a hash that matches the hash in the registry, the custom preference is used. Locate custom preferences under one of the following methods:

- Base 16 encoded SHA1 public key hash of the CA certificate. A SHA1 hash of the public key is used instead of SHA1 hash of the certificate so that the preferences can survive cross certification where the SHA1 hash of the public key remains the same.
- Hex representation of an OID followed by two NULL characters.

Identifying Certificates with a Hash of the Public Key

Adobe's Certificate Viewer provides an easy way to get the public key hash.

To do so:

1. Import the certificate into Acrobat.

2. Open the Trusted Identify Manager.

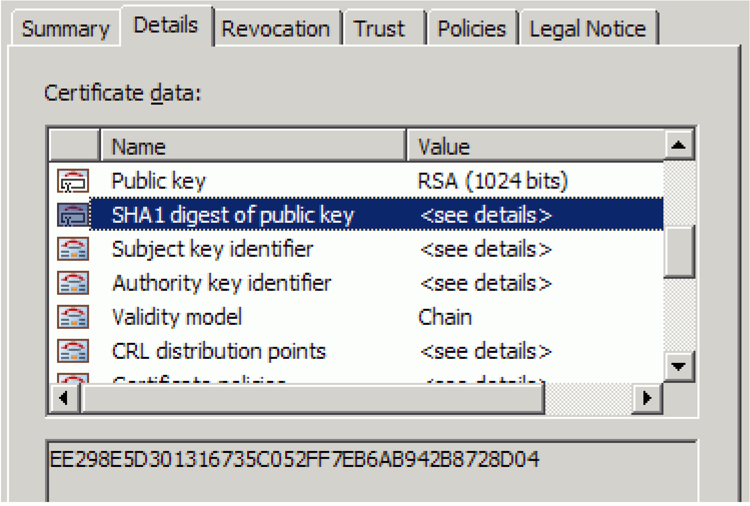
3. Choose **Certificates** from the Display drop down list.

4. Highlight the certificate you will use (an ICA).

5. Choose **Show Certificate**.

6. Choose the Details tab.

Figure 24 Certificate Viewer details tab



Certificate data:

Name	Value
Public key	RSA (1024 bits)
SHA1 digest of public key	<see details>
Subject key identifier	<see details>
Authority key identifier	<see details>
Validity model	Chain
CRL distribution points	<see details>
Certificate policies	<see details>

EE298E5D301316735C052FF7EB6AB942B8728D04

7. Highlight **SHA1 digest of public key**.

8. Copy the hash in the lower panel to the clipboard. This example uses `5D800FA2F49D4816FCA014B B9442665922BA8A77`.

9. Open the registry

10. Navigate to `HKEY_CURRENT_USER\Software\Adobe\<application>\<version>\Security\cASPKI\cASPKI\cCustomCertPrefs\`.

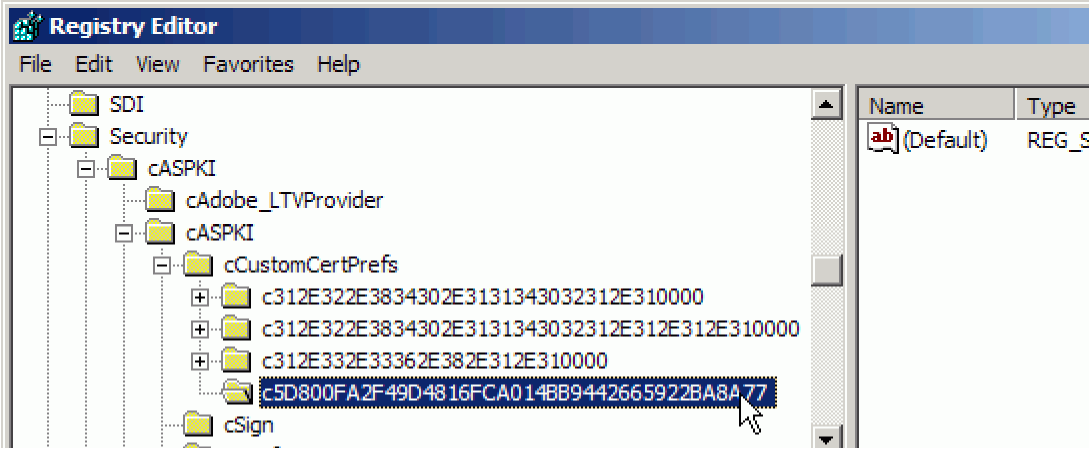
11. Right click on `cCustomCertPrefs`.

12. Choose **New** > **Key**.

13. Enter "c" followed by the public key hash you just created. For example: `c5D800FA2F49D4816FCA014BB9442665922BA8A77`.

This registry key is now ready to be populated with custom preferences and should look like Figure 25.

Figure 25 Custom certificate preference key


Registry Editor
File
Edit
View
Favorites
Help
SDI
Security
cASPKI
cAdobe_LTVProvider
cASPKI
cCustomCertPrefs
c312E322E3834302E3131343032312E310000
c312E322E3834302E3131343032312E312E312E310000
c312E332E33362E382E312E310000
c5D800FA2F49D4816FCA014BB9442665922BA8A77
cSign
Name
Type
(Default)
REG_S

Adding Certificate Preferences

On a Windows machine, certificate-specific preferences can be added by following the steps below:

1. Navigate to `HKEY_CURRENT_USER\Software\Adobe\<application name>\<version number>\Security\cASPKI\cASPKI\cCustomCertPrefs\<your ID hash>`.

2. Add the needed containers and keys. Custom entries under cCustomCertPrefs are always a cab and the name is prepended with a "c." For example, to set a timestamp server provider preferences, you would use the available timestamp preferences. While the key list shows `sPassword` as a valid name, when it is used under `CustomCertPrefs`, the entry should be renamed to `cPassword`.

   Requirements will vary based on your specific need:

   - cAdobe_LTVProvider
   - cAdobe_TSPProvider
   - cAdobe_OCSPRevChecker
   - cAdobe_CRLRevChecker
   - cAdobe_ChainBuilder

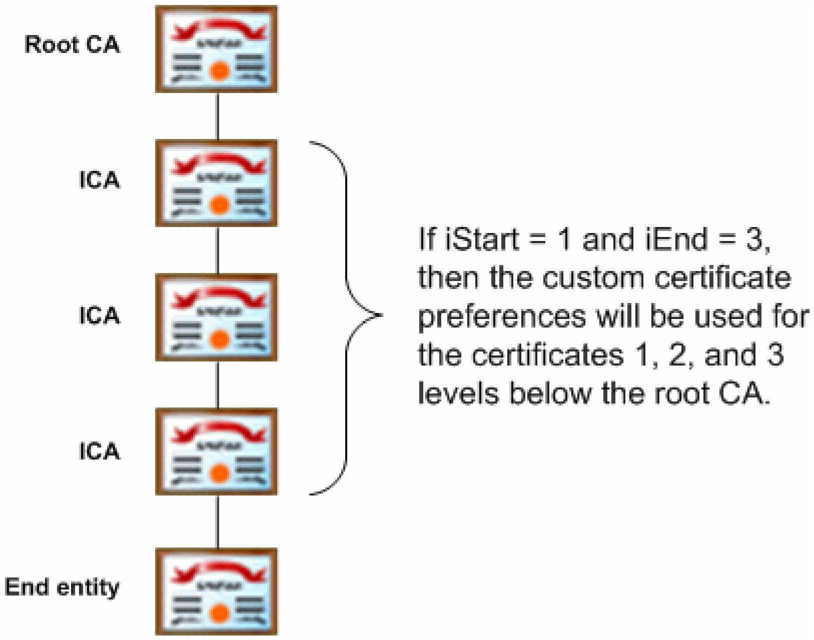
Associating a Certificate Preference with a Chain Scope

`iStart` and `iEnd` can be used to specify for what parts of a certificate chain a custom certificate preference will apply. They are always used at the container level of c0, c1, c2, and so on. For example, Acrobat could be configured to search for acceptable policy OIDS only in the certificates that are the first, second, and third levels below the root CA (Figure 26).

Table 21 Registry preferences: Scoping certificate

Name	Type	Description
iStart	int	(v 7.0) Determines the start of the preference relevance depth relative to the certificate chain. By default, the preference starts at the current level.
iEnd	int	(v 7.0) Determines the end of the preference relevance depth relative to the certificate chain. By default the depth of the preference is MaxUns32

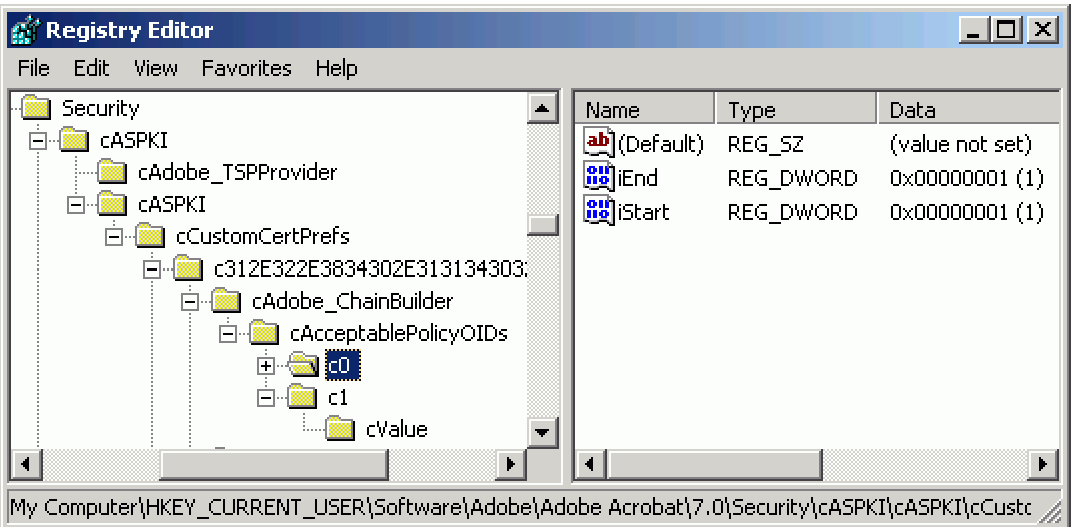
Figure 26 Specifying preference relevance


Root CA
ICA
ICA
ICA
End entity
If iStart = 1 and iEnd = 3, then the custom certificate preferences will be used for the certificates 1, 2, and 3 levels below the root CA.

To specify a scope within a chain:

1. Navigate to `<root>\cCustomCertPrefs\<certificate public key hash>\cAdobe_ChainBuilder\cAcceptablePolicyOIDs\c0`.
2. Highlight c0, right click, and choose **New > DWORD**.
3. Enter the field names `iStart` and `iEnd`.
4. Right click on a field and choose **Modify**.
5. Set the **Value Data** field to specify the needed start or end range.
6. Choose **OK**.
7. Restart the application.

Figure 27 Preference relevance keys



Name	Type	Data
(Default)	REG_SZ	(value not set)
iEnd	REG_DWORD	0x00000001 (1)
iStart	REG_DWORD	0x00000001 (1)

My Computer\HKEY_CURRENT_USER\Software\Adobe\Adobe Acrobat\7.0\Security\cASPKI\cASPKI\cCustc

A Custom Certificate Preference Example for Identrus Compliance

Acrobat has two custom certificate preferences that enable Identrus compliance at `<security root>\cASPKI\cASPKI\cCustomCertPrefs\<hash of Identrus OID>`. You can use this as a template for a similar custom certificate preference:

- c312E322E3834302E3131343032312E310000
- c312E322E3834302E3131343032312E312E312E310000>

Whenever a credential that chains up to an Identrus CA is used for signing or signature validation, the application follows the Identrus rule set rather than the default rule set. The custom certificate preferences enable Acrobat to recognize Identrus certificates and process them as required by Identrus.

When Acrobat starts for the first time, it checks whether `bCustomPrefsCreated` is set. If not set to `true`, Acrobat writes out the Identrus rule set that is hard coded within Acrobat. If it's already set, then writing out the Identrus rules is skipped.

Note: If the Identrus rules in Acrobat need to be changed without updating Acrobat, then create a custom installer (tuned with the wizard) that sets `bCustomPrefsCreated` to `true` and writes out the new Identrus rules. Acrobat will then first check whether custom preferences have been created. If they are already created, then Acrobat won't write out the Identrus rules within Acrobat, and those written out by the custom installer will be respected by ASPKI.

An Identrus CA is identified by the certificate policy OID 1.2.840.114021.1.1.1 present in the production Identrus root CA. However, the certificate policy OID present in their test root CA is different, and in order to be able to test Acrobat against the test Identrus environments, Acrobat also uses the same Identrus rules for CAs that have the certificate policy OID 1.2.840.114021.1. Identrus certificates must contain one of the OIDs listed in AcceptablePolicyOIDs in Table 22. If the OID is not present, the certificate is deemed to be invalid and the signature will also be invalid.

Table 22 Registry preferences: Identrus

Name	Type	Description
cAcceptablePolicyOIDs\c1	cab of strings	(v. 7.0) Default: "iEnd"=dword:00000002 and "iStart"=dword:00000002 The default chain scope in which to look for the policy OIDs.
cAcceptablePolicyOIDs\c1\cValue	cab of strings	(v. 7.0) Default: The values described below. An array of strings containing the policy OIDs for a certificate to be considered acceptable. **For ICA certificates**: Set to 1.2.840.114021.1.6.1 and 1.2.840.114021.1.2.1 **For EE certificates**: Set to 1.2.840.114021.1.4.1, 1.2.840.114021.1.4.2, 1.2.840.114021.1.7.2, 1.2.840.114021.1.10.1, 1.2.840.114021.1.10.2, 1.2.840.114021.1.13.2, 1.2.840.114021.1.16.2, 1.2.840.114021.1.19.2, 1.2.840.114021.1.22.2, 1.2.840.114021.1.25.2, 1.2.840.114021.1.28.2, 1.2.840.114021.1.30.2

6.1.5 Custom Security Handler Preferences

`Adobe.PPKLite` is the default security handler used for performing private key functions, validating signatures, and signing and encrypting documents. This is represented in the user interface as *Adobe Default Security* in the Digital Signatures Advanced Preferences dialog on both the Verification and Creation tabs. Administrators can install custom handlers to perform these functions, in which case the drop down lists on these tabs will list the additional handlers. All entries in the *cHandler* folder are reset by the Digital Signature Preferences dialog's **Reset** button.

Security handlers are Acrobat plugins. Information about creating plugins in general and security handlers in particular can be found in the Acrobat Software Development Kit (SDK) and its HFTs, header files, and other API documentation. Because Acrobat's `Adobe.PPKLite` is becoming more feature rich with each release, it is unlikely that you will need a custom security handler.

If a custom handler is used, you can specify the following:

- Separate handlers for signing/encryption and signature validation.
- The default method displayed in the drop-down list of handlers.
- Lock down the selections so they cannot be modified by end users.

Table 23 Registry preferences: Security handler

Name	Type	Description
aPrivKey	ASAtom	(v 7.0) Default: Adobe.PPKLite **Path**: <security root>\cHandlers Maps to GUI item: **Method to use When Signing and Encrypting Documents** on the Creation tab of the Digital Signature Advanced Preferences dialog. Used by DigSig and PubSec to store the handler that accesses private key functions. It is used for signing, decryption, and responding to an FDF file request to export contact information. The value should be set to *Adobe.NoHandler* if it is desired that the user be asked to select a handler. Subject to lockdown as described in "Preventing End-User Modification" on page 44.

Table 23 Registry preferences: Security handler

Name	Type	Description
aVerify	ASAtom	(v 7.0) Default: Adobe.NoHandler **Path**: <security root>\cHandlers Maps to GUI items: The selected radio button determines the value: • **Use the document-specified method, prompt if it is not available**: Sets the value to Adobe.NoHandler • **Use the document-specified method, use the default method if it is not available**. Sets the value to Adobe.PPKLite • **Always use the default method (overrides the document-specified method)**. Takes the value selected from **Default Method for Verifying Signatures**. (aPrivKey) Remembers the name of the preferred handler to use when verifying signatures. If this value is not set, then the handler used to verify signatures is the handler that matches the `Filter` attribute in the signature dictionary; if this handler is not available, then the user is prompted to select a handler. If this value is set then, its meaning is qualified by the value of `bVerifyUseAlways`. Subject to lockdown as described in "Preventing End-User Modification" on page 44.
bVerifyUseAlways	boolean	(v 7.0) Default: 0 **Path**: <security root>\cHandlers Maps to GUI item: **Always use the default method (overrides the document-specified method)** Qualifies the use of `aVerify`. If true and `aVerify` is set to a handler name, then this handler is used to verify all signatures. If false, then the `aVerify` handler is used only to verify signatures when the handler specified by the signature dictionary `Filter` attribute is not present. Subject to lockdown as described in "Preventing End-User Modification" on page 44.
aAddressBook	ASAtom	(v 7.0) Default: null **Path**: <security root>\cHandlers Remembers a preferred handler for accessing Trusted Identity Manager functions including certificate data import from an FDF file.
aDirectory	ASAtom	(v 7.0) Default: null **Path**: <security root>\cHandlers Remembers a preferred handler for directory functions (e.g. LDAP), including for importing directory information from an FDF data exchange file.
cDialogs: xSelHandler	4 int keys	(v 7.0) Default: null **Path**: <security root>\cDigSig The last on-screen coordinates of a handler's digital ID selection dialog. It is a subkey containing 4 keys: Top, Bottom, Left, and Right. This preference could be used by 3rd party handlers or by someone invoking a non-signing digital ID selection dialog via JavaScript.

6.1.6 ASPKI Service Provider Interfaces

A service provider interface (SPI) is part of an architectural model that provides a programming interface for developing replaceable components and common services access. As a standalone PKI toolkit written in C++, ASPKI has no dependencies on the PDF Library or Acrobat and is designed to be portable and usable in different applications, including but not limited to Acrobat and GUI-less servers. It provides a number of SPIs which may be configured independently of each other.

The SPI preferences in `<security root>\cASPKI\cSPIs` are not exposed to end users and can only be set by an administrator that needs to fine tune the application's certificate handling. For example, a company may not want to use Adobe or MSCAPI as a credential provider, might need to control revocation checking in a custom way, or might wish to prevent the use of PKCS#11 devices. In most deployments, the default behavior should be sufficient and you should not need to modify the settings. However, customization is possible as follows:

- "Revocation Checker Providers" on page 65
- "Signature Validation Directory Providers" on page 66
- "Roaming ID Authentication Mechanisms" on page 67

6.1.6.1 Revocation Checker Providers

The revocation checker provider provides revocation checking services. You can specify one or more revocation checking methods and choose whether to use the default methods or some MSCAPI-specific method.

The following options are available:

- Use one or both of Adobe's revocation checking methods (CRL and OCSP).
- Use of the MSCAPI revocation checking plugin model as an alternative to Adobe mechanisms. For example, administrators may have standardized on MSCAPI or might prefer the MSCAPI method of using a CRL registry cache (Acrobat has its own cache).

Note: Acrobat's default CRL cache location is `C:\Documents and Settings\<user>\Application Data\Adobe\<application>\<version>\Security\CRLCache`

Table 24 Registry preferences: SPI for revocation checker providers

Name	Type	Description
iRevocationChecker	int	(v 7.0) Default: 2 **Path**: <security root>\cASPKI\cSPIs **0**: Use none of the registered providers. **1**: Use first registered provider. **2**: Use all registered providers.

Table 24 Registry preferences: SPI for revocation checker providers

Name	Type	Description
cRevocationChecker	cab	(v 7.0) Default: Adobe_OCSPRevChecker, Adobe_CRLRevChecker **Path**: <security root>\cASPKI\cSPIs An array of text entries (t0-tn) containing the name of a registered provider: **Adobe_OCSPRevChecker**: Adobe's default OCSP method. **Adobe_CRLRevChecker**: Adobe's default CRL method. **MSCAPI_RevocationChecker**: Accesses MSCAPI revocation checking plugin framework. The rules of operation are as follows: ● If cRevocationChecker is empty, the default OCSP and CRL methods are used. ● If cRevocationChecker is not empty, then only the methods listed are used. ● Regardless of the order in which the validators are listed, the validators are always called in the following order: OCSP, CRL, MSCAPI. ● The first validator present that produces a result is the only one used.

6.1.6.2 Signature Validation Directory Providers

The directory provider SPI provides access to trust anchors and intermediate CAs used for signature validation. By default, certificates in all of the supported locations are used.

The following options are available:

- Preventing or allowing access to certificates in P12 files. End users must also be logged in to the file.
- Preventing or allowing access to certificates in the Trusted Identity Manager (AAB).
- Preventing or allowing access to certificates in the Window Certificate Store.
- Preventing or allowing access to self-signed certificates created by an Adobe application.

Table 25 Registry preferences: SPI for directory providers

Name	Type	Description
iDirectoryProvider	int	(v 7.0) Default: 2 **Path**: <security root>\cASPKI\cSPIs **0**: Use none of the registered providers. **1**: Use first registered provider. **2**: Use all registered providers.
cDirectoryProvider	cab	(v 7.0) Default: All **Path**: <security root>\cASPKI\cSPIs An array of text entries (t0-tn) containing the name of a registered provider: **Adobe_FileCredentialDirectoryProvider**: Provides access to PKCS#12 files. **AAB_DirectoryProvider**: Provides access to the Trusted Identity Manager. **MSCAPI_DirectoryProvider**: Provides access to the Windows Certificate Store. **Adobe_SelfSignedCredDirectoryProvider**: Provides access to self signed certificates created by Acrobat.

6.1.6.3 Roaming ID Authentication Mechanisms

The authentication mechanism provider pertains only to roaming IDs. It enables you to specify one or more authentication mechanisms. The mechanism must be supported by the roaming ID server with which the application communicates. The following features are available:

- Enabling multiple authentication mechanisms.
- Limiting the authentication mechanism to one specified type.
- Turning off authentication so that roaming IDs cannot be used.

Table 26 Registry preferences: SPI for authentication mechanisms

Name	Type	Description
iAuthMechanisms	int	(v 8.0) Default: 2 **Path**: <security root>\cASPKI\cSPIs **0**: Use none of the registered providers. **1**: Use first registered provider. **2**: Use all registered providers.
cAuthMechanisms	cab	(v 8.0) Default: The array contains all four values. **Path**: <security root>\cASPKI\cSPIs An array of text entries (t0-tn) where each entry contains the name of a registered provider: **PLAIN**: A mechanism defined in RFC2595 consisting of a single message specifying the user's ID and password. **ASSP-Kerberos**: A mechanism commonly used on Windows that passes a Single Sign On token and receives back a SAML assertion. **ASSP-ArcotID**: A mechanism recognized by Arcot roaming ID servers. **ASSP-QnA**: A mechanism that initiates a question-answer dialog between the user and server.

6.1.7 FDF Import and Export

The File Data Exchange Format (FDF) provides a format for easily importing and exporting certificate data and application settings. These settings appear in **`<security root>\cPubSec`** after a client uses the feature.

The default values are stored internally by the application and are not visible in the registry. An administrator can set the default behavior, but your configuration is subject to modification by end users via the user interface.

The following features are available:

- Specifying whether the default export behavior is to save or email the file.
- Specifying whether the default export behavior is to sign the file.
- Specifying whether the default certificate request behavior is to save or email the file.
- Enabling or disabling WebBuy FDF processing (deprecated).

Example 6.6: FDF preferences

```
[HKEY_CURRENT_USER\Software\Adobe\Adobe Acrobat\8.0\Security\cPubSec]
"bFDFRequestExcludeCert"=dword:00000000
```

```
"bFDFRequestSave"=dword:00000000
```

Table 27 Registry preferences: FDF import and export

Name	Type	Description
bFDFExportSave	bool	(v 7.0) Default: 1 **Path**: <security root>\cPubSec\ Maps to GUI item: **Save as** and **Email** radio buttons in export dialog. Persists whether user chose to save (1) or email (0) the FDF during export.
bFDFExportSign	bool	(v 7.0) Default: 0 **Path**: <security root>\cPubSec\ Persists whether the user chose to sign the FDF during export.
bFDFRequestExcludeCert	bool	(v 7.0) Default: 0 **Path**: <security root>\cPubSec\ Similar to the `bFDFRequestSave`. False includes the user's certificate in all certificate requests. True excludes it.
bFDFRequestSave	bool	(v 7.0) Default: 0 **Path**: <security root>\cPubSec\ When building a request for someone's certificate, the user is asked whether they want to save the request as an FDF or email it directly. This flag is the cached answer to that question.
bWebBuyFDF	bool	(v 7.0) Default: 1 Deprecated. **Path**: <security root>\cPubSec\ Enables WebBuy FDF file processing.

6.1.8 Security Settings Console

Security Settings Console preferences persist information about the state of the console user interface. These preferences are user generated and implementation specific and are likely to change across application versions.

Note: These keys are not customizable and are provided for informational purposes only.

Table 28 Registry preferences: Security Settings Console

Name	Type	Description
iSVS	int	(v. 7.0) Default: null **Path**: <security root>\cSecurityConsole\ Indicates (in pixels) the position of the vertical window splitter.
iSHS	int	(v. 7.0) Default: **Path**: <security root>\cSecurityConsole\ Indicates (in pixels) the position of the horizontal window splitter.

Table 28 Registry preferences: Security Settings Console

Name	Type	Description
xCategory	binary	(v. 7.0) Default: null **Path**: <security root>\cSecurityConsole\ A binary ID of the last-selected category in the tree view.
cOpenCategories	array	(v. 7.0) Default: null **Path**: <security root>\cSecurityConsole\ An array of binary IDs for all categories in the tree view that were opened.

6.1.9 Displaying All Chains in the Certificate Viewer

By default, the Certificate Viewer builds and displays the trusted chain from the EE to the trust anchor. However, it is possible to show all found chains whether they are trusted or not. While most users do not need this information, it can be used for troubleshooting and verification. End users can turn this option on and off by using the Certificate Viewer's checkbox **Show all certification paths found**. The following option is available:

- Showing all certification paths by default.

Table 29 Registry preferences: Signature validation

Name	Type	Description
bShowAllChains	bool	(v 7.0) Default: 0 **Path**: <security root>\cPPKHandler If true, the Certificate Viewer shows all the chains; otherwise, it shows only the trusted chain. If there are no trusted chains, then all the chains are shown and this preference is ignored.

6.1.10 Configuring Password Caching

By default, password caching is turned on so that users will not always have to enter a password when one is required. This feature affects Adobe Policy Server log in, signing with digital IDs in the Acrobat store (pfx or p12 files), changing password timeout policies, and creating new password security policies. For example, setting the option to false disables the menu option **Save password with the policy** when creating a new policy.

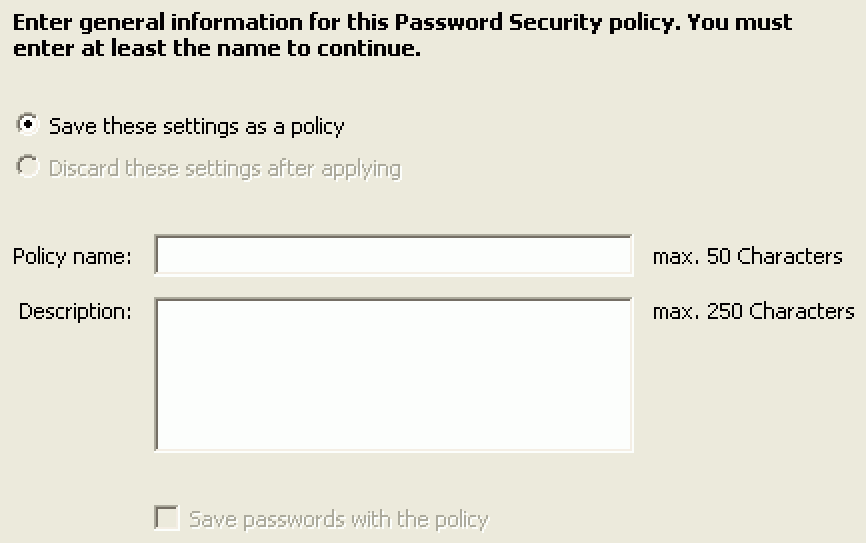
The following options are available:

- Controlling whether some passwords are cached to disk.
- Disabling the option to save a password with a policy.
- Streamlining Adobe LiveCycle Right Management Server workflows. This key does not exist in HKCU. It can only be used in HKLM.

Table 30 Registry preferences: Password caching

	Type	
bAllowPasswordSaving	bool	(v 7.0) Default: 1 **Path**: HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Adobe\<product>\<version>\FeatureLockDown\cSecurity\cPPKLite Maps to GUI item: User interface items where passwords are used: **Save passwords with the policy** in the New Security Policy dialog; **Never** checkbox on the Password timeout dialog. Controls whether certain passwords can be cached to disk. If false, users are prompted to enter a password every time one is required. Not all passwords are affected by this setting.

Figure 28 Password caching: Disabled UI item



Enter general information for this Password Security policy. You must enter at least the name to continue.

● Save these settings as a policy

○ Discard these settings after applying

Policy name: max. 50 Characters

Description: max. 250 Characters

□ Save passwords with the policy

6.1.11 Turning on FIPS Mode

This is a Windows only feature.

Acrobat and Reader can provide encryption via the Federal Information Processing Standard (FIPS) 140-2 mode. FIPS 140 is a cryptographic security standard used by the federal government and others requiring higher degrees of security. Through registry configuration it is possible to force Acrobat to use only FIPS 140-certified cryptographic libraries. Doing so only affects the production and not the consumption of PDF files, and it only affects encryption and digital signature workflows.

When the FIPS mode is on, encryption uses FIPS-approved algorithms provided by the RSA BSAFE as follows:

- v. 9.x and earlier: Crypto-C ME 2.1 encryption module with FIPS 140-2 validation certificate 608.
- v. 10: Crypto-C ME 3.0.0.1 encryption module with FIPS 140-2 validation certificate 1092

FIPS mode changes Acrobat’s default behavior as follows:

- FIPS-compliant algorithms are always used.

- Self-signed certificate creation is disabled. In FIPS mode, users cannot create self-signed certificates.
- Signing with non-FIPS supported algorithms results in an error message; that is, signing fails if the document hash algorithm (digest method) is set to MD5 or RIPEMD160.
- Password security is turned off. Users can apply certificate or Adobe LiveCycle Right Management Server security using the AES encryption algorithm to a document, but password encryption is disabled.
- When applying certificate security, the RC4 encryption algorithm is not allowed.
- Documents protected with non-FIPS compliant algorithms cannot be saved.

Table 31 Registry preferences: FIPS mode

Name	Type	Description
bFIPSMode	bool	(v 8.1) Default: 0 **Path**: HKEY_CURRENT_USER\Software\Adobe\Adobe Acrobat\<version>\AVGeneral Turns FIPS mode on and off.

6.2 Digital Signatures

Signing is a complex process that could involve multiple signatures on a changing document. Moreover, support for multiple types of IDs, revocation checking methods, and so on mean that Acrobat is as feature rich as it is heavy with customizable preferences. For this reason, a basic understanding of the signing and signature validation process is prerequisite to customizing application behavior. Once you are ready to configure the application, refer to the following sections for details:

- “Revocation Checking Preferences” on page 71
- “Signature Creation Preferences” on page 79
- “Signing Workflow Preferences” on page 86
- “Signature Validation Preferences” on page 92
- “Configuring Password Caching” on page 69

Note: Signing and signature validation behavior is also subject to control via general preferences such as the digital ID and service provider interface preferences described in “Setting Basic Client and Workflow Preferences” on page 44.

6.2.1 Revocation Checking Preferences

Since revocation checking can occur both during signature creation and signature validation, revocation settings may affect both the user’s ability to sign and to validate signatures. A check can occur for the signing certificate as well as for the certificates associated with any revocation check responses. OCSP and CRL checking are both supported, and MSCAPI checks will use one or the other or both.

Interaction between OCSP and CRL Preferences

If a signature has the following chain CA | ICA | EE and OCSP revocation checking preferences are specified for the CA but CRL preferences are not, then `Adobe_OCSPRevChecker` and `Adobe_CRLChecker` behave as follows:

- While doing OCSP revocation checking for the ICA, the preferences specified for the CA are used. If the scope for the preferences is specified as infinite, then the CA preferences are also used for revocation checking the EE.
- While doing CRL revocation checking, if no preferences have been specified either for the CA or the ICA, then the preferences present at `cASPKI:cAdobe_CRLRevChecker` are used instead.

Tip: For more detail about how revocation checking affects signing and signature validation, see the Digital Signatures Addendum.

The following options are available:

- "Configuring OCSP Revocation Checking" on page 72
- "Configuring CRL Revocation Checking" on page 76
- "Certificate Chain Building" on page 78

6.2.1.1 Configuring OCSP Revocation Checking

OCSP revocation checking can occur both during signature creation and signature validation on both the signing certificate as well as for the certificates associated with any revocation check responses. The following options are available:

- "Specifying the Time and Method of OCSP Checks" on page 73
- "Specifying Certificates for Valid OCSP Responses" on page 75
- "Specifying Certificates for Valid OCSP Requests" on page 76

Tip: For more detail about how revocation checking affects signing and signature validation, see the Digital Signatures Addendum..

OCSP responders: Determining if they are authorized to do rev checks

RFC2560 defines three methods of determining whether the OCSP responder is authorized to perform OCSP revocation checking. Two methods are strictly defined and the third one is called "local configuration" which Acrobat defines by specifying a set of certificates. If OCSP response is signed with one of these certificates then the responder is considered authorized.

Rule 1 (Acrobat 8.0): defined the local configuration rule as follows by authorizing OCSP responses that come from responders specified by sURL.

Rule 2(Acrobat 9.0): If a custom certificate preference has a new "AuthorizedResponder" boolean entry with a value of true, and the certificate being checked for OCSP revocation as well as the OCSP response both chain up to the customer certificate, then the responder is authorized.

The order in which verifying OCSP responders occurs is as follows:

1. Local configuration rule #1 (A8 and A9).

2. Local configuration rule #2 (new in A9).

3. The two deterministic methods from RFC2560.

> **Note:** The structure and location of the new AuthorizedResponder entry is the same as for SendNonce entry. However, while SendNonce may be specified under ASPKI or a custom certificate preference, AuthorizedResponder may be specified only under custom certificate preferences.

Specifying the Time and Method of OCSP Checks

OCSP revocation checking preferences allow you to control when and how an OCSP check occurs. The following options are available:

- Specifying when to do revocation checking as well as the effect of a failed or bad response.
- Specifying when and where to go online to get a response.
- Specifying whether to include a nonce. Nonces are random generated numbers that are sent with a revocation check request and matched by a response. They improve security by assuring communication with an active, non-spoofed server.
- Using or ignoring a response's `thisUpdate` and `nextUpdate` times to control its validity. See RFC 2560 for details.
- Setting a limit on the amount of time difference between the local time and response's publish time.

OCSP response embedding changes with 10.1

Prior to 10.1, OCSP responses without nextUpdate were never embedded in a signature. For 10.1 and later, OCSP responses are always embedded irrespective of the presence of nextUpdate; however, whether they are used for signature validation depends on certain conditions:

- Validation time is greater than thisUpdate minus the value of `maxClockSkew` (the default is 5 minutes). This test is always performed.
- When nextUpdate is present and the validation time is less than the nextUpdate time plus the value of `maxClockSkew`.
- When nextUpdate is not present and the validation time is less than the thisUpdate time or the producedAt time (whichever is greater) plus the value of `iMaxClockSkew`.

If you need a relaxed security environment (for example, when the responder is caching OCSP responses), `bIgnoreNextUpdate` can be set to 1 to ignore the last test. In this case, embedded responses without nextUpdate are always used for signature validation provided that they pass first test.

> **Note:** This behavior is designed to support Acrobat's long term validation feature and allows validating a signature with embedded responses that were valid at signing time.

Table 32 Registry preferences: OCSP method

Name	Type	Description
iReqRevCheck	int	(v 7.0) Default: 2 **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker Indicates whether revocation checks are required to succeed on the OCSP response. Interacts with other iReqRevCheck settings as described in the Digital Signatures Addendum.. **0**: Don't do revocation checks. **1**: Do a check IF certificate has AIA extension or responder info is in registry; don't fail if the check fails. **2**: Do a check IF certificate has AIA extension or responder info is in registry; all checks must succeed if there is data and a check occurs. **3**: Require a check; it must succeed under all circumstances.
iURLToConsult	int	(v 7.0) Default: 0 **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker Specifies how the revocation checker chooses which responder to use: **0**: Use the AIA extension in the certificate. **1**: Use the URL key in `sURL`. **2**: Use the AIA extension in the certificate. If it is not present, use the URL key in `sURL`. **3**: Use the OCSP request signer's certificate AIA extension. Relevant only if **`SignRequest`** is 0.
sURL	string	(v 7.0) Default: null **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker The URL used to fetch OCSP responses.
bGoOnline	bool	(v 7.0) Default: 1 **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker Specifies whether to go online to do revocation checking. Never used for Reader enabled signatures (UR3).
bSendNonce	bool	(v 7.0) Default: 1 Deprecated with 10.0. See iSendNonce. **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker If true, nonces are included in the OCSP request and expected to be present in the response and should match the request's nonce. If false, nonces are not sent.
iSendNonce	int	(v 10.0) Default: 2 **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker With 10.0, this preference replaces bSendNonce. Possible values include: **0**: No nonces are sent. **1**: Nonces are included in the OCSP request and expected to be present in the response and should match the request's nonce. **2**: Nonces are included in the OCSP request, but if none are present in the response, their abscence is ignored.

Table 32 Registry preferences: OCSP method

Name	Type	Description
iResponseFreshness	int	(v 7.0) Default: 525600 (1 year) **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker Specifies the amount of time in minutes after the response's published `thisUpdate` time for which the response will be valid. After that time, the response will be invalid.
bIgnoreValidityDates	bool	(v 7.0) Default: 0 **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker Specifies whether to the response's thisUpdate and nextUpdate times, thereby preventing any negative affect of these times on response validity. The value is set to true for ubiquity signatures created by enabling usage rights for Adobe Reader.
iMaxClockSkew	int	(v 7.0) Default: 5 minutes **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker The number of minutes the local machine time can vary from the response's published time to account for a network delay, time synchronization issues, and so on. Since 10.1, this setting is also used to determine whether embedded OCSP responses should be used to validate signatures in conjunction with the nextUpdate and thisUpdate. For details, see "OCSP responders: Determining if they are authorized to do rev checks" on page 72.
bIgnoreNextUpdate	bool	(v 10.1) Default: 0 **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker For 10.1 and later, this preference is used along with iMaxClockSkew to determine whether or not embedded OCSP responses are actually used for signature validation. For details, see the subfeature description above. This behavior is designed to support Acrobat's long term validation feature and allows validating a signature with embedded responses that were valid at signing time.Possible values include: • 0: iMaxClockSkew is applied to thisUpdate on both sides of the validation time, i.e. thisUpdate - iMaxClockSkew < validation time < checkTime + iMaxClockSkew where checkTime is the later of the producedAt and thisUpdate. When true, iMaxClockSkew is applied to thisUpdate only before the validation time: thisUpdate - iMaxClockSkew < validation time. • 1: If there is no nextUpdate, then we accept the OCSP response indefinitely (we do not check for if the validation time is too late) and don't check whether validation time is < than checkTime)

Specifying Certificates for Valid OCSP Responses

It is possible to require certain features for certificates used to sign OCSP responses. If a response does not meet the specified parameters, it is considered invalid and the signature status may be Unknown or Invalid. The following options are available:

- Allowing or disallowing the `OCSPNoCheck` extension.
- Requiring the presence of a public key hash extension (`bRequireOCSPCertHash`).

Table 33 Registry preferences: OCSP certificate

Name	Type	Description
bAllowOCSPNoCheck	bool	(v. 8.0) Default: 1 **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker Specifies whether the OCSPNoCheck extension is allowed in the response signing certificate.
bRequireOCSPCertHash	bool	(v. 8.0) Default: 0 **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker Specifies whether a certificate public key hash extension must be present in OCSP responses.

Specifying Certificates for Valid OCSP Requests

It is possible to require certain features for certificates used to sign OCSP requests. If a request does not meet the specified parameters, it is considered invalid and the signature status may be Unknown or Invalid. The following options are available:

- Specifying whether OCSP requests should by signed (`bSignRequest`).
- Requiring the presence of a particular OID in a request (`sSignCertOID`).

Table 34 Registry preferences: OCSP certificate

Name	Type	Description
bSignRequest	bool	(v 7.0) Default: 0 **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker Specifies whether outgoing OCSP requests should be signed.
sSignCertOID	string	(v 7.0) Default: null **Path**: <security root>\cASPKI\cAdobe_OCSPRevChecker A certificate policy OID prefix required in the signing certificate. When set, Acrobat looks at all the available digital IDs to see if one contains the requisite prefix. Only used if bSignRequest is true.

6.2.1.2 Configuring CRL Revocation Checking

CRL revocation checking can occur both during signature creation and signature validation on both the signing certificate as well as for the certificates associated with any revocation check responses. The following options are available:

- "Specifying the Time and Method of CRL Checks" on page 76
- "Specifying Certificates for Valid CRL Checks" on page 78

Tip: For more detail about how revocation checking affects signing and signature validation, see the Digital Signatures Addendum..

Specifying the Time and Method of CRL Checks

CRL revocation checking preferences allow you to control when and how a CRL check occurs. The following options are available:

- Specifying when to do revocation checking as well as the affect of a failed or bad response.
- Specifying when and where to go online to get a response.
- Setting a time limit for caching a response after which the application must get a new response.
- Specifying a LDAP server to query for CRLs.

Note: Querying an LDAP server can result in poor application performance depending on the quality of the network connection and the number of directories to search.

Table 35 Registry preferences: CRL

Name	Type	Description
iReqRevCheck	int	(v 7.0) Default: 1 **Path**: <security root>\cASPKI\cAdobe_CRLRevChecker Indicates whether revocation checks are required to succeed on the CRL response. Interacts with other iReqRevCheck settings as described in the Digital Signatures Addendum.. **0**: Don't do revocation checks. **1**: Do a check IF responder details are in CRLDp certificate extension or the registry; don't fail if the check fails. **2**: Do a check IF responder details are in CRLDp certificate extension or the registry; all checks must succeed if there is data and a check occurs. **3**: Require a check; it must succeed under all circumstances.
sURL	string	(v. 7.0) Default: null **Path**: <security root>\cASPKI\cAdobe_CRLRevChecker\cURLDP The URL used to fetch CRL responses for an additional URL CRL Distribution point.
bAlwaysConsult	bool	(v. 7.0) Default: 0 **Path**: <security root>\cASPKI\cAdobe_CRLRevChecker\cURLDP Optional. Determines when the URL is used for an additional URL CRL distribution point. If false, the URL is only used when the certificate does not have a CRLDp extension.
bGoOnline	bool	(v. 7.0) Default: 1 **Path**: <security root>\cASPKI\cAdobe_CRLRevChecker Indicates whether it's acceptable to go online to fetch a CRL. If false, only cached CRLs (on local disk or ones embedded with signature) are consulted. Internally set to false for ubiquity signatures in Reader enabled documents internally.
sLDAP	string	(v. 7.0) Default: Nulll **Path**: <security root>\cASPKI\cAdobe_CRLRevChecker The LDAP server to get CRLs from in the form www.ldap.com. Without the protocol prefix, as LDAP is assumed. All DN-based queries for CRLs will be directed to this server.
iMaxRevokeInfoCacheLifetime	int	(v 7.0) Default: 24 (hours) **Path**: <security root>\cPubSec Maximum cache lifetime of the information (e.g. CRL) used to do revocation checking.

Specifying Certificates for Valid CRL Checks

It is possible to require certain features for certificates used to sign CRL responses. If a response does not meet the specified parameters, the response will be considered invalid and the signature status may be Unknown or Invalid. The following options are available:

- Specifying whether to ignore the response certificate's times in the `thisUpdate` and `nextUpdate` extensions.
- Requiring the presence of the Authority Key Identifier extension.

Table 36 Registry preferences: CRL

Name	Type	Description
bIgnoreValidityDates	bool	(v. 7.0) Default: 0 **Path**: <security root>\cASPKI\cAdobe_CRLRevChecker Specifies whether to ignore the response's `thisUpdate` and `nextUpdate` times, thereby preventing any negative affect of these times on response validity. The value is set to true for ubiquity signatures created by enabling usage rights for Adobe Reader.
bRequireAKI	bool	(v. 8.0) Default: 0 **Path**: <security root>\cASPKI\cAdobe_CRLRevChecker Specifies whether the Authority Key Identifier extension must be present in a CRL.

6.2.1.3 Certificate Chain Building

The revocation checking process includes building the certificate chain so that each discovered certificate can be analyzed and processed as specified by other application preferences. Administrators do have some control over what certificates are used to build a chain. The following options are available:

- Controlling whether AIA extensions are followed.
- Requiring the use of valid RSA signatures on all certificates in a chain.
- Requiring the presence of specific policy OIDs in the specified chain scope for it to be valid.
- Pointing to an LDAP server for path discovery purposes. Querying an LDAP server can result in poor application performance depending on the quality of the network connection and the number of directories to search.

Table 37 Registry preferences: Chain building

Name	Type	Description
bFollowURIsFromAIA	bool	(v. 7.0) Default: 0 **Path**: <security root>\cASPKI\cAdobe_ChainBuilder If true, the chain builder is allowed to follow URIs in AIA certificate extensions so that certificates can be downloaded if they are not available locally. The default does not allow phone-home capability. CRLdps and OCSP AIA extensions do allow following URIs because they require that the certificate chain up to a trust anchor.

Table 37 Registry preferences: Chain building

Name	Type	Description
bRequireValidSigForChaining	bool	(v. 8.0) Default: 0 **Path**: <security root>\cASPKI\cAdobe_ChainBuilder If true, the chain builder will not build chains with invalid RSA signatures on certificates. Consider chain CA > ICA > EE where the CA's signature on an ICA is invalid. If this setting is true, the chain building will stop at the ICA and the CA will not be included in the chain. If this preference is false, the full 3-certificate chain is produced. This setting does not affect DSA signatures.
cValue	cab of strings	(v. 7.0) Default: null **Path**: <security root>\cASPKI\cAdobe_ChainBuilder\cAcceptablePolicyOIDs\c<index> An array of strings c0-cN containing the required certificate policy OIDs. Note that c<index> can be associated with a chain scope as described in "Associating a Certificate Preference with a Chain Scope" on page 60.
sLDAP	string	(v. 7.0) Default: null **Path**: <security root>\cASPKI\cAdobe_ChainBuilder If present, specifies the URL of an LDAP server to be used for path discovery.

6.2.2 Signature Creation Preferences

Tip: For a one page guide, see "Signature Creation Workflow" on page 99.

Signature creation preferences can control the type and strength of signature as well as what conditions will cause signing to fail. The available options allow you to configure signing workflows for varying degrees of security (Figure A.1). The following options are available:

- Preferences that control the signature strength and format:
  - "Changing the Default Hashing Algorithm" on page 79
  - "Specifying a Signing Format" on page 80
- Preferences that control what is included in a signature:
  - "Embedding Revocation Data in Signatures" on page 82
  - "Setting up Timestamp Servers" on page 84
- Preferences that control when to abort the signature creation process:
  - "Requiring a Pre-Signing Digest Comparison" on page 83
  - "Requiring Signature Property Retrieval" on page 83

6.2.2.1 Changing the Default Hashing Algorithm

The default algorithm used to create a message digest (document hash) during signing can be customized. In some enterprise situations, such as when FIPS compliance is required, you may need a more secure algorithm. Alternate hashing algorithms can be specified by name or OID as shown below. The algorithm used is displayed in the **Hash Algorithm** field of the Signature Property dialog Document tab.

Usage rules:

- MSCAPI supports different algorithms across versions. For example, early XP versions only supported SHA1 and MD5. The use of other algorithms will require that the signer use a digital ID that resides in a .pfx/.p12 file in the Acrobat cache.
- With XP SP3, MSCAPI supports SHA256 on certificates and some token devices.
- Pre 9.1: Acrobat uses SHA1 as the default.
- 9.1 and later: Acrobat uses SHA256 as the default, but will use SHA1 if the token does not support SHA256.
- If using FIPS mode, do not use MD5 or RIPEMD160.

The following options are available:

- Specifying an alternate algorithm.

Table 38 Registry preferences: Signing algorithm

Name	Type	Description
aSignHash	ASAtom	(v 7.0) Default: SHA1 for 9.0 and earlier; SHA256 for 9.1 and later **Path**: <security root>\cPubSec The hashing algorithm to use while signing. The ASAtom is a binary entry that uses the name of a supported algorithm: ● MD5 ● RIPEMD160 ● SHA1 ● SHA256 **(v. 8.0)** ● SHA384 **(v. 8.0)** ● SHA512 **(v. 8.0)** For an alternative, see tSignHash.
tSignHash	ASAtom	(v 7.0) Default: SHA1 for 9.0 and earlier; SHA256 for 9.1 and later **Path**: <security root>\cPubSec A text entry that contains the OID of the hashing algorithm: ● 1.2.840.113549.2.5: MD5 ● 1.3.36.3.2.1: RIPEMD160 ● 1.3.14.3.2.26: SHA1 ● 2.16.840.1.101.3.4.2.1: SHA256 **(v. 8.0)** ● 2.16.840.1.101.3.4.2.2: SHA384 **(v. 8.0)** ● 2.16.840.1.101.3.4.2.3: SHA512 **(v. 8.0)** For an alternative, see aSignHash.

6.2.2.2 Specifying a Signing Format

The default format for creating the signature object that is embedded in a signed document is PKCS#7. The object contains the encrypted message digest, certificates, timestamps, and other information. It does not include the signature appearance and data outside of `Contents` in the signature dictionary. Format choices are limited so that a signature encoded by one handler can be unencoded (validated) by another handler. Providing a value for `aSignFormat` writes that value to the signature dictionary's `SubFilter` object. For details, see "Signature Interoperability" in the *PDF Reference*.

- **PKCS#1**: For signing PDF files using PKCS#1, the only recommended value of `SubFilter` is adbe.x509.rsa_sha1, which uses the RSA encryption algorithm and SHA-1 digest method. The certificate chain of the signer is stored in the `Cert` entry.
- **PKCS#7:** The value of `Contents` is a DER-encoded PKCS#7 binary data object containing the signature. The PKCS#7 object must conform to the PKCS#7 specification in Internet RFC 2315, PKCS #7: Cryptographic Message Syntax, Version 1.5. `SubFilter` can take one of the following values:
  - **adbe.pkcs7.detached**: No data is encapsulated in the PKCS#7 signed-data field.
  - **adbe.pkcs7.sha1**: The SHA1 digest of the byte range is encapsulated in the PKCS#7 signed-data field with `ContentInfo` of type `Data`.
- **ETSI.CAdES.detached**: Supports long term validation of signatures even when the signing certificate is revoked; this is part of the feature which allows adding an invisible timestamp signature to a document.

Table 39 Registry preferences: Signing format

Name	Type	Description
aSignFormat	ASAtom	(v 7.0) Default: adbe.pkcs7.detached **Path**: <security root>\cPubSec Maps to GUI item: **Preferences > Security > Advanced Preferences > Creation tab > Default Signature Signing Format** The format to use when signing a document using public key cryptography when a format is not specified by a seed value, javascript parameter, or the PubSec Handler. Allowable values include: • adbe.pkcs7.detached • adbe.pkcs7.sha1 • adbe.x509.rsa_sha1 • ETSI.CAdES.detached

6.2.2.3 Configuring Revocation Checking for Signing

Applying a signature to a document involves both creating a signature and then validating it. Despite the fact that end users see only one step (the signature appears with a status icon), there are actually two phases which an administrator can configure independently of the other. Revocation checking can occur during the initial signing phase to control whether or not a signature is created. The following option is available:

- Specifying when to do revocation checking as well as the affect of a failed or bad response.

Note: Interacts with `bIsEnabled`. For more detail about how revocation checking affects signing and signature validation, see the Digital Signatures Addendum..

Table 40 Registry preferences: Revocation checking (signing)

Name	Type	Description
iReqRevCheck	int	(v 7.0) Default: 0 **Path**: <security root>\cASPKI\cASPKI\cSign Indicates whether revocation checks are required to succeed to create the signature. Interacts with other iReqRevCheck settings as described in the Digital Signatures Addendum.. **0**: Don't do revocation checks. **1**: Do a check IF CRLDp or AIA information resides in the certificate or registry; don't fail if the check fails. **2**: Do a check IF CRLDp or AIA information resides in the certificate or registry; all checks must succeed if there is data and a check occurs. **3**: Require a check; it must succeed under all circumstances.

6.2.2.4 Embedding Revocation Data in Signatures

Whether revocation checking information is stored in a signature varies by version. Storing such data in a signature enables offline revocation checking and a determination of whether a signer's certificate was valid at the time of signing.

Setting `bIsEnabled` to 1 via the GUI or registry automatically sets `cSign\iReqRevCheck` to 2. The rationale is that if you choose to embed the revocation status you probably want a status to embed. A consequence of this choice is that you must do a check and retrieve a good result; otherwise, no signature is created. In other words, signing with a revoked certificate is prevented when this setting is on.

The following options are available:

- Embedding revocation status in a signature.
- Specifying the embedded data cache size to limit the amount of cached data.

Tip: If you are setting up a signing workflow for both signers and signature validators, you may want to set `iUseArchivedRevInfo` so that document recipients will validate signatures based on the signer's `bIsEnabled` setting.

Table 41 Registry preferences: Long term validation

Name	Type	Description
bIsEnabled	bool	(v. 8.0) Default: Pre 9.1 0; 9.1 and later: 1 **Path**: <security root>\cASPKI\cAdobe_LTVProvider **Include signature's revocation status when signing** Specifies whether the signature revocation status is included in the signature.
iMaxRevInfoArchiveSize	int	(v. 7.0) Default: 10K **Path**: <security root>\cASPKI\cAdobe_LTVProvider The maximum size of the revocation archival info in kilobytes. An attempt is made to store as much revocation information as possible without exceeding the limit.

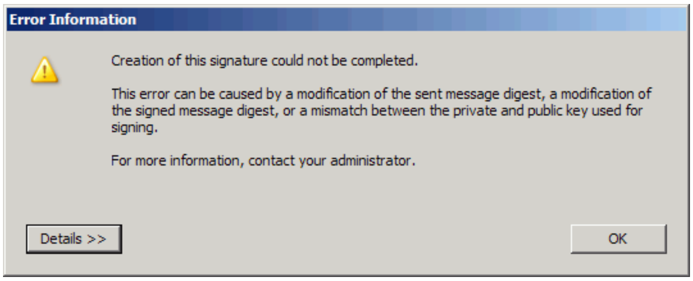
6.2.2.5 Requiring a Pre-Signing Digest Comparison

When signing a PDF document, a message digest is created for the document and sent to the cryptographic module that performs the signing operation. Setting the registry entry `bEnforceSecureChannel` to 1 ensures the message digest sent to the cryptographic module is checked against the signed message digest that it returns. This flag ensures that intermediate layers of software between Acrobat and the cryptographic module do not tamper with the signing operation.

> **Note:** When using a certificate that includes a DSA public key with omitted parameters, the test to detect signature validity is not performed. In these cases, setting `bEnforceSecureChannel` has no effect.

When this preference is turned on, a digest mismatch results in a warning dialog (Figure 29). The signature is removed from the document and the signing application aborts the signing process.

Figure 29 Secure channel error



Creation of this signature could not be completed.

This error can be caused by a modification of the sent message digest, a modification of the signed message digest, or a mismatch between the private and public key used for signing.

For more information, contact your administrator.

Table 42 Registry preferences: Message digest comparison

Name	Type	Description
bEnforceSecureChannel	bool	(v 8.0) Default: 0 **Path**: <security root>\cPubSec This setting prevents signing when the original message digest and the signed message digest do not match. This error can be caused by a modification of the original message digest, a modification of the signed message digest, or a mismatch between the private and public key used for signing. When this preference is on, the user sees a warning dialog when the digest mismatch occurs. When using a certificate that doesn't include a public key (such as a DSA certificate with an omitted public key), the test to detect signature validity is not performed. Do not turn this setting on if such certificates are used.

6.2.2.6 Requiring Signature Property Retrieval

Acrobat currently provides a signature property for timestamps. By default, retrieving a valid and trusted timestamp is not required, and property retrieval failure only results in creating a signature which uses the local time.

The following option is available:

- Requiring property retrieval in order to create a signature. If fetching a timestamp fails for any reason (bad URL, no network connection, etc.) the signature creation process is aborted and no signature is created.

Table 43 Registry preferences: Signature property retrieval

Name	Type	Description
bReqSigPropRetrieval	bool	(v 7.0) Default: 0 **Path**: <security root>\cASPKI\cASPKI\cSign Indicates whether retrieving a signature property must succeed. For example, if a user configures a bad timestamp server URL and makes it the default, a signature is not created. An error appears. **0**: Make best effort, but success is not required. A signature is created. **1**: Property retrieval must succeed. On failure, a signature is not created and an error dialog appears.

6.2.2.7 Setting up Timestamp Servers

Timestamp servers are automatically used during signing when a server is configured and selected as a default. The default server is identified by a star in the Security Settings Console. `cPPKHandler` can be used to specify a list of servers and `cAdobe_TSPProvider` can specify the default server that is actually used during signing.

End users can overwrite preconfigured values by editing server settings in the Security Settings Console. Note that if a user sets a new default server, the console values will overwrite the values in `cAdobe_TSPProvider`. Values under `cPPKHandler` do not get written to `cAdobe_TSPProvider` unless that server is selected as the default.

The following options are available:

- Specifying a list of servers that will appear in the Security Settings Console. Preferences are represented as a list c0-cN and contain the server name, URL, and whether the authentication is required (Table 44).
- Setting a default server (Table 45).
- Specifying when to do revocation checking as well as the affect of a failed or bad response.
- Increasing security by choosing a more robust hashing algorithm. The algorithm must be supported by the timestamp server.
- Requiring signature property retrieval (a valid and trusted server URL) in order to create a signature. For details, see "Requiring Signature Property Retrieval" on page 83.

Note: See also "Signature Validation Preferences" on page 92.

Table 44 Registry preferences: Timestamp server list

Name	Type	Description
tServer	cab	(v 7.0) Default: null **Path**: <security root>\cPPKHandler\cTimeStampServers\c<index> The server URL. Describes a timestamp server. It is created from the Security Settings dialog with the timestamp Server configuration.

Table 44 Registry preferences: Timestamp server list

Name	Type	Description
tName	text	(v 7.0) Default: null **Path**: <security root>\cPPKHandler\cTimeStampServers\c<index> The user-defined server name. This can be Unicode.
bAuthRequired	bool	(v 7.0) Default: null **Path**: <security root>\cPPKHandler\cTimeStampServers\c<index> Maps to GUI item: **This server requires me to log on** If true, indicates that the above timestamp server requires authentication.
xLockboxId	string	(v 7.0) Default: null **Path**: <security root>\cPPKHandler\cTimeStampServers\c<index> Maps to GUI item: The preference is populated when the user checks **This server requires me to log** on and then enters a username and password. If a timestamp server requires authentication, the authentication data is stored in a secure store such as Microsafe and is identified by this ID. The service provider needs to know what type of secure store the identifier names. Only used when ASPKI is running within the Acrobat environment.

Table 45 Registry preferences: Default timestamp server

Name	Type	Description
iReqRevCheck	int	(v 7.0) Default: 2 **Path**: <security root>\cASPKI\cAdobe_TSPProvider Indicates whether revocation checks on timestamps are required to succeed before signing. Failure does not affect signature creation or validation, it only results in defaulting to the local, machine time. Interacts with other iReqRevCheck settings as described in the Digital Signatures Addendum.. The possible values include: **0**: Don't do revocation checks. **1**: Do a check IF CRLDp or AIA information resides in the certificate or registry; don't fail if the check fails. **2**: Do a check IF CRLDp or AIA information resides in the certificate or registry; all checks must succeed if there is data and a check occurs. **3**: Require a check; it must succeed under all circumstances.
sURL	string	(v 7.0) Default: null **Path**: <security root>\cASPKI\cAdobe_TSPProvider A timestamp server URL such as http://www.example.com/tsp. Because no default is specified, it must be configured for timestamping to work. Only the HTTP(s) protocol is supported.
bAuthRequired	bool	(v 7.0) Default: null **Path**: <security root>\cASPKI\cAdobe_TSPProvider If true, indicates that the above timestamp server requires authentication.

Table 45 Registry preferences: Default timestamp server

Name	Type	Description
sUser	string	(v 7.0) Default: null **Path**: <security root>\cASPKI\cAdobe_TSPProvider The server login username. Relevant only if `bAuthRequired` is true. Only username and password-based authentication is supported.
sPassword	string	(v 7.0) Default: null **Path**: <security root>\cASPKI\cAdobe_TSPProvider The server log in password. Relevant only if bAuthRequired is true.
xLockboxId	string	(v 7.0) Default: null **Path**: <security root>\cASPKI\cAdobe_TSPProvider Maps to GUI item: The preference is populated when the user checks **This server requires me to log on** and then enters a username and password. If a timestamp server requires authentication, the authentication data is stored in a secure store such as Microsafe and is identified by this ID. The service provider needs know what type of secure store the identifier names. Only used when ASPKI is running within the Acrobat environment.
iHashAlgo	int	(v 7.0) Default: 1 **Path**: <security root>\cASPKI\cAdobe_TSPProvider Identifies the hashing algorithm used to hash the timestamped data. The valid values are: **0**: MD5 **1**: SHA1 (Default prior to 9.1) **2**: SHA256 (Supported beginning with 8.0 and the default since 9.1) For an alternative, see sHashAlgo which supports more options.
sHashAlgo	string	(v 8.0) Default: SHA1 **Path**: <security root>\cASPKI\cAdobe_TSPProvider The hashing algorithm OID used to hash the data to be timestamped. The valid values are: • **MD5**: 1.2.840.113549.2.5 • **SHA1**:1.3.14.3.2.26 • **SHA256**: 2.16.840.1.101.3.4.2.1 • **SHA384**: 2.16.840.1.101.3.4.2.2 • **SHA512**: 2.16.840.1.101.3.4.2.3
iSize	int	Default: (v 7.0) 4096 (v. 10.0) 6144 **Path**: <security root>\cASPKI\cAdobe_TSPProvider ASPKI requires the signature property to predict the size (in bytes) so that enough space can be set aside.

6.2.3 Signing Workflow Preferences

Signing workflow preferences control what end users can see and do when they invoke the signing dialog. You can require certain actions, hide and display data fields, and change how the signing process is affected by content which might impact the users ability to see what they are signing. The following options are available:

- “Requiring Preview Mode to Sign” on page 87
- “Specifying Signature Appearances” on page 87

- "Preconfiguring Location and Contact Information" on page 88
- "Specifying Signing Reasons" on page 88
- "Certification Signature Preferences" on page 89
- "Controlling Signature Warnings: Review" on page 90
- "Controlling Signature Warnings: Fonts" on page 91

6.2.3.1 Requiring Preview Mode to Sign

Preview mode turns off (suppresses) content and dynamic behavior that could prevent the signer from seeing what they are signing. While the use of preview mode adds an extra step in the signing workflow, it turns off potentially bad content, checks the document for the presence of any PDF constructs that may cause problems with signature integrity, and provides a report about any found problems. For more information about preview mode, see Chapter 4, "Document Integrity and Preview Mode".'

The following option is available:

- Force the use of preview mode during signing.

Table 46 Registry preferences: Preview mode

Name	Type	Description
bPreviewModeBefore Signing	bool	(v 7.0) Default: 0 **Path**: <security root>\cDigSig Maps to GUI item: **Preferences > Security > View documents in preview mode when signing** Specifies whether a signer is forced to use preview mode during signing. If true, preview mode is automatically invoked on a sign action. Users should read the document message bar text, view a report about any warnings, and then choose **Sign Document**.

6.2.3.2 Specifying Signature Appearances

The application remembers what signature appearance a signer used and stores its index number in `iAPIndex`. Because an end user's appearance selection will overwrite any custom value here, customization by an administrator would server no useful purpose.

Note: This key is not customizable and is provided for informational purposes only.

Table 47 Registry preferences: Appearance

Name	Type	Description
iAPIndex	int	(v 7.0) Default: null. **Path**: <security root>\cPubSec Remembers the last used signature appearance index.

6.2.3.3 Preconfiguring Location and Contact Information

The signing dialog has the capability of showing a location and contact information fields during a signing workflow. Field fill-in is optional. By default, the option is off, but end users and administrators can turn this option on. The location will appear in the Signature Properties dialog and in the Signature's pane and may optionally appear in the signature appearance. The following options are available:

- Showing or not showing the **Contact** and **Location** fields in the signing dialog.
- Setting default contact information.
- Setting default location information.

Note: If the end user changes the field data in the signing dialog, those values will overwrite the registry-specified values.

Table 48 Registry preferences: Signing information

Name	Type	Description
bAllowOtherInfoWhenSigning	bool	(v 8.0) Default: 0 **Path**: <security root>\cPubSec Maps to GUI item: **Preferences > Security > Advanced Preferences > Creation tab > Show location and contact information when signing** Specifies whether the location and contact information UI will appear during signing.
tContactInfo	text	(v 7.0) Default: null **Path**: <security root>\cPubSec Maps to GUI item: Contact field in the Sign dialog. When bAllowOtherInfoWhenSigning is true (on), the signing dialog displays a location and contact field. User data is saved and reused during subsequent signing events.
tLocation	text	(v 7.0) Default: null **Path**: <security root>\cPubSec Maps to GUI item: Location field in the Sign dialog. When bAllowOtherInfoWhenSigning is true (on), the signing dialog displays a location and contact field. User data is saved and reused during subsequent signing events.

6.2.3.4 Specifying Signing Reasons

The signing dialog has the capability of showing a signing reasons drop down list during a signing workflow. By default, the option is off, but end users and administrators can turn this option on. If a reason is used, it appears in the signature appearance, the Signature Properties dialog, and in the Signatures pane. The following options are available:

- Showing or not showing the **Reasons** field in the signing dialog.
- Changing the default reasons. Administrators can add, delete, and modify the reason list.
- Locking the reason list so that it can't be modified by end users.

Table 49 Registry preferences: Signing reason

Name	Type	Description
bAllowReasonWhenSigning	bool	(v 8.0) Default: 0 **Path**: <security root>\cPubSec Maps to GUI item: **Preferences > Security > Advanced Preferences > Creation tab > Show reasons when signing** Specifies whether the reason UI will appear during signing. The preference can be overridden by a document seed value set on a field. 8.1: Subject to lockdown as described in "Preventing End-User Modification" on page 44. If cReasons is locked and is empty, bAllowSigningReasons is 0 and read only (The UI is turned off). If cReasons is locked and has values, then bAllowSigningReasons is true and read only.
cReasons	text	(v 7.0) Default: See below. **Path**: <security root>\cPubSec Maps to GUI item: **Reasons drop down list in signing dialog** List of reasons for signing. Subject to override by the document seed value: `reasons`. Entries in this folder are named t0, t1, etc. The default reasons are: **t0**: I am the author of this document **t1**: I have reviewed this document **t2**: I am approving this document **t3**: I attest to the accuracy and integrity of this document **t4**: I agree to the terms defined by the placement of my signature on this document **t5**: I agree to specified portions of this document Subject to lockdown as described in "Preventing End-User Modification" on page 44.

6.2.3.5 Certification Signature Preferences

These preferences only control certification signature behavior and have no effect on approval signature behavior. In addition to the general signature preferences described elsewhere in this document, the following options are available:

- **Preventing invisible signatures**: By default, users can sign with a visible or invisible signature. Prohibit invisible certification signatures by setting `bAllowInvisibleSig` to 0.
- **Legal attestations (warning comments)**: When certifying a document that contains dynamic content, a signer can choose a default warning comment from a list or create a custom one. You can prepopulate this list with custom comments with `cAttest`.
- **(Pre v. 8.0) Control certification based on document content**: For versions prior to 8.0, you can control certification rights based on the nature of the document content and whether it generates LegalPDF warnings. These preferences are deprecated in 8.0.

Table 50 Registry preferences: Certification signature

Name	Type	Description
bAllowInvisibleSig	bool	(v 7.0) Default: 1 **Path**: <security root>\cDigSig Maps to GUI item: Certify with Invisible Signature If true, invisible signatures are allowed. False disables the menu option, prevents signing and certifying with invisible signatures, and limits JavaScript support by signature fields. Subject to lockdown as described in "Preventing End-User Modification" on page 44.
cAttest	cab of text	(v 7.0) Default: null **Path**: <security root>\cDigSig List of most recently used attestations regarding LegalPDF warnings in a document. Entries in this folder are named t0, t1, etc. The application may have one or more default strings such as "I have included this content to make the document more interactive."
bAllowCertNonGreen	bool	(v 7.0 ONLY) Default: 1 **Path**: <security root>\cDigSig If true, a certification signature may be applied to a document containing Legal PDF warnings. If false, then its not allowed and the author is informed of the reason.
bAllowSigCertOnly	bool	(v 7.0 ONLY) Default: 0 **Path**: <security root>\cDigSig Specifies whether any subsequent signers can sign a certified document containing LegalPDF warnings with additional approval signatures. In other words, the presence of any LegalPDF warning prevent any additonal signatures.
bAllowSigCertGreenOnly	bool	(v 7.0 ONLY) Default: 0 **Path**: <security root>\cDigSig Specifies whether any subsequent signers can sign a certified document that does not contain LegalPDF warnings with additional approval signatures. In other words, the presence of any LegalPDF warning does not prevent any additonal signatures.

6.2.3.6 Controlling Signature Warnings: Review

The Sign dialog is capable of showing a **Review** button. The button invokes the PDF Signature Report which analyzes the document for the presence of any dynamic content that could adversely affect the integrity of signing workflows. If none is found, a dialog appears indicating that there are no problems. If content such as a comment or JavaScript is discovered, the PDF Signature Report appears with a list of any PDF constructs that may cause problems with signature integrity. The following options are available:

- Never showing or allowing the review of document warnings.
- Limiting warning review to certification workflows.
- Requiring warning review prior to applying an approval and/or certification signature.
- Always requiring review of warnings for every signature.

Table 51 Registry preferences: Document warning

Name	Type	Description
iShowDocumentWarnings	int	(v 8.0) Default: 1 **Path**: <security root>\cPubSec Maps to GUI item: **Preferences > Security > Advanced Preferences > Creation tab > Enable reviewing of document warnings** Specifies whether a button to allow reviewing document warnings shows up on the signing dialog. Interacts with `iRequireDocumentWarnings`. **0**: Never **1**: Show when certifying only **2**: Always
iRequireReviewWarnings	int	(v 8.0) Default: 0 **Path**: <security root>\cPubSec Maps to GUI item: **Preferences > Security > Advanced Preferences > Creation tab > Prevent signing until document warnings are reviewed**. Specifies whether the user is required to review document warnings before signing via the signing dialog. Interacts with `iShowDocumentWarnings`. **0**: Never **1**: Show when certifying only **2**: Always

6.2.3.7 Controlling Signature Warnings: Fonts

LegalPDF warnings have been replaced by PDF Signature Report errors in versions 8.0 and later. Both mechanisms provide similar warnings. The following option is available:

- Toggling warnings for true type and non-embedded fonts on and off.

Table 52 Registry preferences: Font warning

Name	Type	Description
bEnNonEmbFontLegPDFWarn	bool	(v 7.0) Default: 0 **Path**: <security root>\cDigSig Turns on and off warnings about non-embedded fonts. A warning appears when the `LegalPDF` dictionary `NonEmbeddedFonts` attribute has a non zero value. Turning this value on causes a warning to appear in the PDF Signature Report which indicates the document contains unembedded fonts.
bTrueTypeFontPDFSigQWarn	bool	(v 7.0) Default: 0 **Path**: <security root>\cDigSig Turns on and off warnings about True Type fonts.

6.2.4 Signature Validation Preferences

Signature validation preferences control the display of status icons, logging, how validation occurs, and so on. Many of these preferences interact with the signing preferences and should be set accordingly. The following options are available:

- "Setting Validation Preferences that Map to UI" on page 92
- "Controlling Signature Status Icon Behavior" on page 94
- "Logging Certificate Validation Data" on page 95
- "Using Embedded Validation Data" on page 96
- "Revocation Checking Constraints" on page 97
- "Displaying All Chains in the Certificate Viewer" on page 69

Tip: For more detail about how revocation checking affects signing and signature validation, see the Digital Signatures Addendum..

6.2.4.1 Setting Validation Preferences that Map to UI

While users can configure general application signature validation preferences via the GUI, admins can preconfigure the application. The following options are available:

- Controlling whether all signatures are validated when a document opens.
- Specifying which time to use when validating a signature.
- Specifying when to do revocation checking as well as the affect of a failed or bad response.
- Using expired timestamps.
- Showing timestamp warnings in the Document Message Bar.

Table 53 Signature validation preferences

Name	Type	Description
bValidateOnOpen	bool	(v 7.0) Default: 1 **Path**: <security root>\cDigSig Maps to GUI item: **Preferences > Security > Verify signatures when the document is opened** Specifies whether to automatically validate all signatures on document open. Subject to lockdown as described in "Preventing End-User Modification" on page 44.
iSigVerificationTime	int	(v 7.0) Default: 1 (2 as of 9.1) **Path**: <security root>\cPPKHandler Maps to GUI item: **Preferences > Security > Advanced Preferences > Verification tab > Verify Time** Each radio button corresponds to a value below: Indicates the time at which signature validation should occur: **0**: Always carry out the verification at current time **1**: Use the signing time if it's secure (e.g. timestamped), else use current time **2**: Always use signing time

Table 53 Signature validation preferences

Name	Type	Description
bUseExpiredTimestamps	bool	(v 9.1) Default: 1 **Path**: <security root>\cAdobe_TSPProvider Maps to GUI item: **Preferences > Security > Advanced Preferences > Verification tab > Use expired timestamps** Specifies whether expired timestamps should be used. If true, an expired timestamp will not invalidate a signature.
iReqRevCheck	int	(v 7.0) Default: 2 **Path**: <security root>\cASPKI\cASPKI\cVerify Maps to GUI item: **Preferences > Security > Advanced Preferences > Verification tab > Require certificate revocation checking to succeed . . .** Indicates whether revocation checks are required to succeed. The user interface exposes this preference as a binary value to simplify the end users experience. A checked checkbox translates to 2 (RequiredIfInfoAvailable). An unchecked checkbox translates to 0 (No checks). This check doesn't affect ubiquity signature verification where the value is always 1. Interacts with other iReqRevCheck settings as described in the Digital Signatures Addendum.. **0**: Don't do revocation checks. **1**: Do a check IF CRLDp or AIA information resides in the certificate or registry; don't fail if the check fails. **2**: Do a check IF CRLDp or AIA information resides in the certificate or registry; all checks must succeed if there is data and a check occurs. **3**: Require a check; it must succeed under all circumstances. Subject to lockdown as described in "Preventing End-User Modification" on page 44.

6.2.4.2 Certificate Chain Validation Method

The application uses shell validation by default, but chain validation may be used when required. Compliance with the German signature law requires chain validation.

Table 54 Registry preferences: Chain validation method

Key	Type	Description
iValidityModel	int	(v. 8.0) Default: 0 **Path**: `<security root>\cASPKI\` Specifies the validity model for validating signatures and certificates. **0**: PKIX shell model **1**: Chain validity model. Chain validation is used to validate all or part of a certificate chain when any certificate chaining up to a CA certificate containing the qualified certificate policy extension (OID 1.3.36.8.1.1) or the validity model certificate extension OID (1.3.6.1.4.1.8301.3.5) with the value set to the chain model OID (1.3.6.1.4.1.8301.3.5.1).

6.2.4.3 Controlling Signature Status Icon Behavior

By default, when an application validates a signature it displays a signature status icon in the Signature Properties dialog, and in the Signatures Pane. You can customize status icon behavior for a particular enterprise requirement. For example, a blue i appears on a signature status icon based on certain rules when a document is changed after it was signed.

The following options are available:

- Turning on the icon for signature appearances with `bSigAPStatusIconDisable`. This is off by default because displaying the signature status within the document represents a security vulnerability.
- Turning off the icon for signature appearances AND remove the Hide signature field validity icon when signature is valid from the user interface so the user cannot change the setting with `iDisplayValidIcon`.
- Turning on the icon for valid signatures only with `iDisplayValidIcon`.
- Turning off the blue i in the Signature Properties dialog, and Signatures Pane with `bShowWarningForChanges`.

Table 55 Registry preferences: Signature status icon

Name	Type	Description
bSigAPStatusIconDisable	bool	(v 7.0) Default: 0 **Path**: <security root>\cDigSig Controls whether the signature status icon is displayed in the signature appearance on the document. If true, status icon is not displayed regardless of signature status. This setting overrides iDisplayValidIcon and bShowWarningForChanges.
iDisplayValidIcon	int	(v 7.0) Default: 2 for versions 9.0 and later. 0 for older versions. **Path**: <security root>\cPubSec Maps to GUI item in versions prior to 9.x: **Preferences > Security > Advanced Preferences > Verification tab > Hide signature field validity icon when signature is valid**. This UI item was removed from versions 9.x and later because signature status was moved to the Document Message Bar. Determines when the signature status icon is displayed in a signature appearance: **0**: Always **1**: Display except when the signature is valid **2**: Never. This value disables bShowWarningForChanges and removes the **Hide signature field validity icon** option from the GUI. This setting does not affect the icons in the Signatures Pane or in the Signature Properties dialog

Table 55 Registry preferences: Signature status icon

Name	Type	Description
bShowWarningForChanges	bool	(v 7.0) Default: 1 **Path**: <security root>\cPubSec Determines whether or not to show a blue i con validated signature(s) if the document changes after it was signed. If true, a document change results in a blue i status icon appearing for validated approval signatures. Use this setting when users need to know a document has changed after it was signed. If false, the status icon remains a green check and pen even if a document changes after it is signed. The setting provides a method for administrators to turn off the blue i in workflows where documents can be changed or signed multiple times. This setting does not affect certification signatures. The warning icon never appears for valid certification or approval signatures in certified documents if the signatures were allowed by the certifier. Interacts with iDisplayValidIcon which cannot be set to 2, or the icons will not appear regardless of how bShowWarningForChanges is set.

6.2.4.4 Logging Certificate Validation Data

Versions 8.x and later enable logging certificate validation and revocation checking information. You can set both the logging level and log location (Table 56). The path must already exist for logging to take place. The following options are available:

- Specifying a logging path and filename.
- Setting a logging level.

Example 6.7: Chain building log file settings

```
[HKEY_CURRENT_USER\Software\Adobe\Adobe Acrobat\8.0\Security\cASPKI\cAdobe_
ChainBuilder]
"iLogLevel"=dword:00000008
"sLogFilePath"=<BINARY path to existing directory for log file>
```

Example 6.8: Log file for troubleshooting certificate validation

```
20070207000213Z: --------------------------
     20070207000213Z: Chain builder: Starting chain validation. Chain length = 3
          20070207000213Z: Processing Certificate: DN: ou=VeriSign Trust 
Network, ou=(c) 1998 VeriSign, Inc. - For authorized use only, ou=Class 2 Public 
Primary Certification Authority - G2, o=VeriSign, Inc., c=US Serial: 
00B92F60CC889FA17A4609B85B706C8AAF
          20070207000213Z: verification time = 20070207000213Z
          20070207000213Z: Processing Certificate: DN: cn=Enterprise Services 
CA, ou=Class 2 OnSite Individual Subscriber CA, ou=Terms of use at https://www.
verisign.com/rpa (c)01, ou=VeriSign Trust Network, o=Adobe Systems Incorporated 
Serial: 0C0DB7043D0427BEB15AECA02DC95903
          20070207000213Z: verification time = 20070207000213Z
          20070207000213Z: Processing Certificate: DN: email=example@adobe.com, 
cn=Ben Writer, ou=Adobe CPS - http://www.adobe.com/misc/CPS.html, ou=www.
verisign.com/repository/CPS Incorp. by Ref.,LIAB.LTD(c)99 Serial: 
5C41B5256825491A4981D4FABFCCA044
          20070207000213Z: verification time = 20070207000213Z
     20070207000213Z: Finished Chain Validation.  TroubleFlags: 0
```

Table 56 Registry preferences: Logging

Name	Type	Description
sLogFilePath	bin	(v. 8.0) Default: null **Path**: <security root>\cASPKI\cAdobe_ChainBuilder Specifies the full path of the text log file. For example: C:\ASPKI.log. The parent folder must already exist.
iLogLevel	int	(v. 8.0) Default: null **Path**: <security root>\cASPKI\cAdobe_ChainBuilder Specifies the log level during chain building and validation. The supported levels are: **1**: fatal errors **2**: possible errors **4**: informational messages **8**: verbose information **f**: all messages

6.2.4.5 Using Embedded Validation Data

Administrators can control how embedded revocation information is used. The following options are available:

- Specifying when archived revocation data is used for revocation checking.

  Tip: If you are setting up a signing workflow for both signers and signature validators, you may want to set `iUseArchivedRevInfo` so that document recipients will validate signatures based on the signer's `bIsEnabled` setting.

- Controlling whether or not revocation data is stored in a JavaScript object.

Table 57 Registry preferences: Signature validation

Name	Type	Description
iUseArchivedRevInfo	int	(v. 7.0) Default: 2 **Path**: <security root>\cASPKI\cAdobe_LTVProvider Indicates whether the revocation information archived with the signature is used for revocation checking. **0**: Never **1**: Use only if more recent info is not available. **2**: Always
bReturnRevInfoToUser	bool	(v 7.0) Default: 0 **Path**: <security root>\cPPKHandler If true, the revocation information is maintained within the `SignatureInfo` object and can be retrieved through JavaScript. For more information, see the *Acrobat JavaScript Reference*.

6.2.4.6 Revocation Checking Constraints

Signature validation can have dependencies on other keys. The following options are available:

- Requiring signature property verification such as timestamps. Signatures will not be valid if this key is true and timestamp verification does not succeed.
- Limiting the number of nested verification sessions to prevent looping.
- Limiting the amount of time the signing time can be after the validation time.

Table 58 Registry preferences: Revocation checking constraints

Name	Type	Description
bReqSigPropVerification	bool	(v 7.0) Defaults: 0 **Path**: <security root>\cASPKI\cASPKI\cVerify Indicates whether signature property verification must succeed or not. If it is required and fails, the signature is not validated. As of 8.0, the only property used is the timestamp URL.
iMaxVerifySession	int	(v8.0) Default: 5 **Path**: <security root>\cASPKI\cASPKI\cVerify Indicates the maximum number of nested verification session allowed. This is used to prevent the application from going into infinite loop verifying the OCSP and/or CRL signer certificates caused by incorrect OCSP and/or CRL certificate setup.
iMaxClockSkew	int	(v 7.0) Default: 65 (minutes) **Path**: <security root>\cPubSec The maximum difference in minutes the signing time is allowed to be after the validation time for which the signature can still be valid. PubSec verifies that a document isn't signed in the future by looking at the verifier's system time and the time embedded in the signature dictionary. Whenever time comes into picture, there is always the possibility that the signer and verifier's times are out of sync. `MaxClockSkew` accommodates such differences.

6.2.5 Document Status Dialog (deprecated)

Note: These preferences are not in the security directory and are deprecated after 7.x.

For application versions prior to 8.0, these preferences control whether the application displays the document status dialog when a user opens a certified document. The dialog contains a **Do not show this dialog next time this document is opened** checkbox allows users to turn the dialog off for the currently viewed document when it is next opened.

The application stores a list of document IDs, and each ID is associated with the user's choice and the last certifying signature status. When the application matches an opened document with an ID in the registry, The application respects the choice as long as the certifying signature's validity status has not changed since the last time the document was opened.

Administrators can turn off the document status dialog for specific documents by distributing a preference which contains the signature status and suppress status for particular document IDs.

Table 59 Registry preferences: Document status dialog

Name	Type	Description
sSuppressListOrder	string	(v 7.0) Default: null **Path**: DocumentStatus An indexe of the most recently accessed documents. This is initialized as the string <abc...zABC...z>. As certified documents are opened, an entry for them is created in cSuppressStatusDocList keyed by the first letter in this string. That letter is then moved to the end of this string (bcd...zABC...za).
cSuppressStatusDocList	cab	(v 7.0) Default: null **Path**: DocumentStatus Contains a list of entries for previously opened certified documents. Each entry here is a subkey of the form c<some letter (a-zA-Z)>), and each subkey contains iLastStatus and xDocID.
iLastStatus	int	(v 7.0) Default: null **Path**: DocumentStatus The certifying signature's status the last time a document was opened.
xDocID	binary	(v 7.0) Default: null **Path**: DocumentStatus An application generated binary ID which is used to identify the document.

6.2.6 Examine Document

The Examine Document dialog box identifies hidden document information that might pose a risk to the integrity of security and signature workflows. Found content is listed and linked to in the Examine Document pane. Users can click on a link to view the content and check/uncheck items to mark them for removal. Checked items are removed when the user selects the **Remove** button. The following options are available:

- Examining a document each time it is closed.
- Examining a document each time it is emailed.

Table 60 Registry preferences: Examine document

Name	Type	Description
bAutoLaunchAtDocClose	int	(v 9.0) Default: 0 **Path**: <security root> Maps to GUI item: **Preferences > Document > Examine document when closing document** Automatically examines the document for hidden content when it is closed.
bAutoLaunchAtSendMail	int	(v 9.0) Default: 0 **Path**: <security root> Maps to GUI item: **Preferences > Document > Examine document when sending document by email** Automatically examines the document for hidden content when it is sent in an email.

8 Supported Standards and RFCs

Acrobat's features adhere to widely accepted standards as listed in Table 7.

Table 7 Standards support

Reference	Feature
PDF Reference 1.7 (ISO 32000-1) http://www.adobe.com/devnet/pdf/pdf_reference.html. See also PDF for Archive (PDF/A) and PDF for Exchange (PDF/X) at http://www.iso.org.	Representing signatures in the PDF language.
RFC 3280, Internet X.509v3 Public Key Infrastructure, Certificate and Certificate Revocation List (CRL) Profile. http://www.ietf.org	CRL revocation checking, chain building, path validation, cross certificates, multiple chains.
RFC 2560, X.509 Internet PKI Online Certificate Status Protocol-OCSP. http://www.ietf.org	OCSP revocation checking.
RFC 3161, Internet X.509 Public Key Infrastructure Time-Stamp Protocol http://www.ietf.org	Timestamping: signing and signature validation.
RFC 3281, Attribute Certificate Profile, S. Farrell, R.Housley April 2002. http://www.ietf.org	Attribute certificates.
RFC 2437, PKCS #1: RSA Cryptography Specifications Version 2.0 (1024, 2048, 4096). http://www.ietf.org	A format used for creating a digital signature object which is embedded in a document.
RFC 2898, PKCS #5: Password-Based Cryptography Specification Ver. 2.0. http://www.ietf.org	Password security.
RFC 2315, PKCS #7: Cryptographic Message Syntax, Version 1.5. http://www.ietf.org	A format used for creating a digital signature object which is embedded in a document.
PKCS #11: URI Scheme http://www.ietf.org	Cryptographic token interface (smart cards, tokens, etc.)
RFC 1321, The MD5 Message-Digest Algorithm http://www.ietf.org	Creating a document hash during signing.
RFC 3174, US Secure Hash Algorithm 1 (SHA1) http://www.ietf.org	Creating a document hash during signing.
FIPS PUB 186-2, Digital Signature Standard, describes DSA signatures. http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf	Digital signatures.
FIPS PUB 197, Advanced Encryption Standard (AES 128, 256). http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf	Certificate security.
ISIS-MTT Specification v.1.1 March 2004. http://www.isis-mtt.org/index.php?id=460&L=1	Attribute certificates.
NIST PKITS "Public Key Interoperability Public Key Interoperability Test Suite Certification Path Validation"	Chain building and path validation, including cross certificates and multiple chains.
OIDS. ASN.1	Object identifiers (OIDs)

Table 7 Standards support

Reference	Feature
RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax. http://www.ietf.org	All.
RFC 2595, Using TLS with IMAP, POP3 and ACAP. http://www.ietf.org	The PLAIN authentication mechanism used by the roaming ID feature.
RFC 3778, The application/pdf Media Type. Adobe Systems Incorporated.	Describes PDF media type, digital signature and encryption
ETSI 102 778 PDF Advanced Electronic Signatures (PAdES), Parts 1,2,3 and 4. http://pda.etsi.org/pda/queryform.asp	Digital signature; especially LTV.
ETSI/ESI Technical Standard (TS) 102 778 http://pda.etsi.org/pda/queryform.asp	Digital signatures.
JITC: Joint Interoperability Test Command PKI compliance test suite	DoD-mandated PKI test suite. Compliant since 7.x. See http://blogs.adobe.com/security/tag/jitc.

Table 8 Support for APIs, organizations, etc.

MSCAPI	Microsoft's CryptoAPI
Keychain	Macintosh's CryptoAPI
Esign	A U.S. law that both Acrobat and EchoSign signatures conform to.

Glossary of Security Feature Terms

Table 8 provides a comprehensive list of security terms and acronyms used in the fields of digital signatures and document security.

Table 8 Security Terms

.apf	See Adobe Profile Files.
.cer	A Microsoft format for digital IDs often stored in the Windows Certificate Store. These IDs can be used by Windows programs as well as the Acrobat product family.
.p12	See PKCS#12.
.p7b	See PKCS#7.
.p7c	See PKCS#7.
.pfx	See PKCS#12.
AATL	See Adobe Approved Trust List
AC	See attribute certificate.
Acrobat's Public Key Infrastructure Library	A standalone PKI toolkit written in C++ with the intention of being completely portable and usable in different applications, including but not limited to, Acrobat and GUI-less servers. ASPKI supports RFC 3280 and NIST compliant chain building and path validation, including support for cross certificates and multiple chains; multiple revocation protocols like CRL (RFC3280) and OCSP (RFC2560); time stamping (RFC3161); and embedded revocation information along with a signature to achieve signature archival.
AdES	See advanced electronic signatures
Adobe Approved Trust List	An Adobe program designed to facilitate trust in PDF signatures by downloading a list of trusted, high assurance root and ICA certificates to Acrobat and Reader v9.0 and above.
Adobe LiveCycle Rights Management Server	Adobe LiveCycle Enterprise Suite (ES) is a SOA J2EE-based (Java 2 Enterprise Edition) server software product from Adobe Systems Incorporated used to build applications that automate a broad range of business processes for enterprises and government agencies.
Adobe Policy Server	As of Acrobat 9, Adobe Policy Server is renamed to Adobe LiveCycle Rights Management Server.
Adobe Profile Files	Adobe's legacy certificate format not used after Acrobat 5. The certificates are stored in . apf files. This format is not supported as of version 9.0.
advanced electronic signatures	A type of electronic signature described in the European Union Signature Directive. Differentiated from a Qualified Electronic Signature in that it may not use a QEC.
AIA	See authority information access.
AIIM	See Association for Information and Image Management.
ALCRMS	See Adobe LiveCycle Rights Management Server.
approval signature	A signature used to indicate approval of, or consent on, the document terms. Acrobat and Reader recognize both approval and certification signatures.
APS	See Adobe Policy Server.

Table 8 Security Terms

ASPKI	See Acrobat's Public Key Infrastructure Library.
Association for Information and Image Management	AIIM is a non profit community that provides education, research, and best practices to help organizations find, control, and optimize their information as well as understand the challenges associated with managing documents and business processes.
attribute certificate	A file that contains the supplemental attributes and extension that is bound to a PKC, but does not itself contain any key data.
authority information access	An extension that is part of a PKC which contains information on how to access either PKC's of issuing certificate(s) (least used) or where to access OCSP revocation information which is what this extension is primarily used for.
basic constraint	An extension within a public key certificate that defines whether or not the certificate has been issued to a CA.
CA	See certificate authority (CA).
CAdES	See Cryptographic Message System Advanced Electronic Signatures.
CAPI	See MSCAPI.
CDS	See Certified Document Services (CDS).
CDS digital ID	A digital ID issued by a Certified Document Services provider.
CDS digital ID certificate	See CDS digital ID.
CEN	European Committee for Standardization.
certificate	That part of a digital ID that contains the public key. Certificates are shared among participants of signature and certificate security workflows in order to verify participant identities.
certificate authority (CA)	An entity which issues digital certificates for use by other parties. It is an example of a trusted third party. CAs are characteristic of many PKI schemes.
certificate revocation list (CRL)	CRL is a method that public key infrastructures use to maintain access to cached or networked lists of unexpired but revoked certificates. The list specifies revoked certificates, the reasons for revocation (optional), and the certificate issue date and issuing entities. Each list contains a proposed date for the next release. Acrobat's CRL revocation checker adheres to RFC 3280 and NIST PKITS except for delta CRLs.
certification signature	A digital signature applied using an individual digital ID or organizational digital ID for the purpose of establishing the authenticity of a document and the integrity of a document's content, including its appearance and business logic.
certified document	A document to which a certification signature has been applied.
Certified Document Services (CDS)	An Adobe program where commercial CA's create a subordinate or ICA below that chains to the Adobe Root certificate. As the Adobe Root is automatically trusted by Acrobat and Reader v6.0 and above, signatures made with credentials that chain to it are also similarly trusted.
certify or certifying	The act of applying a certification signature to a document using the Acrobat "Certify" feature. Certifying helps establish document authenticity as well as the integrity of its content, including its appearance and business logic.
Click thru signature	A type of electronic signature where the signer is indicating their agreement with terms and indication to sign the document / process by clicking on a button, which might say "I accept." Typically, an audit log of the event is kept for evidentiary purposes and authentication may or may not be required. Sometimes a server applied digital (certification) signature may be applied after the click thru process to protect the integrity of the document.
CMS	See Cryptographic Message Syntax.

Table 8 Security Terms

CRL	See certificate revocation list (CRL).
Cryptographic Message Syntax	A syntax is used to digitally sign, digest, authenticate, or encrypt arbitrary message content which is described in RFC 3369.
Cryptographic Message System Advanced Electronic Signatures	An EU Advanced Signature Format relying on CMS signatures, as described in ETSI TS 101 733.
cryptographic service provider	Application software that allows it to use MSCAPI to communicate with cryptographic module APIs such as PKCS#11 modules, PFX files, and so on
CSP	See cryptographic service provider.
CSP	Certificate Service Provider. Alternate term for CA.
digest method	A hash algorithm used to create a one way hash of data. Acrobat supports six digest methods; MD5, SHA1, SHA256, SHA384, SHA 512, and RIPEMD 160.
digital ID	An electronic representation of data based on the ITU-T X.509 v3 standard, associated with a person or entity. It is often stored in a password-protected file on a computer or network, a USB token, a smart card, or other security hardware device. It can be used for digital signatures and certificate security. "Digital ID" is sometimes used interchangeably with "certificate"; however, a certificate is only one part of a digital ID which also contains a private key and other data.
digital signature	An electronic signature that can be used to verify the identity of the signer through the use of public key infrastructure (PKI) technology. Signers need a digital ID and an application capable of creating a signature.
digital signature algorithm	An encryption algorithm used to create a digital signature created by NIST as defined by FIPS 186.
digitally sign	To apply a digital signature using a digital ID.
document integrity	In signing workflows, document integrity refers to whether or not what was signed has changed after signing. That is, what the signer signed should be reproducible and viewable on the document recipient's end. For the document recipient to validate a signature, its important to determine to what document or what document version that signature applies. See message digest.
DSA	See digital signature algorithm.
EC	European Commission.
EE	See end entity certificate (EE).
EESSI	Electronic Exchange of Social Security Information.
eID	See electronic ID.
electronic ID	A broad term for any electronic ID. In the EU, it is commonly used to describe national-level identity cards.
electronic signature	Generic term. Generally defined as an electronic process which intrinsically links some tag (data, voice, image, key) to content that is being signed, is linked to the signer, and is generally capable of detecting changes in the document signed. A digital signature is a type of electronic signature, as are click thru and signature image.
embedded JavaScript	JavaScript that exists within a document rather than that which is executed from the JavaScript Console or through a batch process.

Table 8 Security Terms

embedded validation response	Information from the digital ID issuer that was used to apply the digital signature and that indicates if the digital ID was valid when the signature was applied. If the digital ID was valid and no one has tampered with the document, the signature will have a status of VALID. Once the digital ID expires or is revoked, it won't be possible to determine if the signature was valid at the time it was applied unless there is an embedded revocation response.
end entity certificate (EE)	The last element of a signing chain. By definition, an end entity certificate does not contain the basic constraint value CA, and is issued to an individual or a pseudo entity (e. g. a department or organization).
ETSI	European Telecommunications Standards Institute.
ETSI/ESI	ETSI/Electronic Signature & Infrastructure Technical Committee.
EU	European Union.
EU Experts Group	A generic term used widely within the EU. Of particular interest is the EUEG on Electronic Procedures.
EU Signature Directive	Directive 1999/93/EC on a Community Framework for Electronic Signatures. Established a Framework for Electronic Signatures and Standards in the European Union.
FIPS	Federal Information Processing Standards: These are publicly announced standards developed by the United States Federal government for use by all non military government agencies and by government contractors. Many FIPS standards are modified versions of standards used in the wider community (ANSI, IEEE, IOS, etc.).
FIPS 140	Federal Information Processing Standard 140: Standard which defines increasing levels of assurance for hardware and software based devices and applications for storing private keys. Level 1 is the lowest assurance, and level 4 is the highest, requiring devices to self-destruct and 'zeroize' themselves if they are compromised in any way.
hardware security module	While actually a generic term for any hardware device designed to securely store digital IDs, HSMs in common parlance are rack mounted servers or hardened PCI cards which are designed for higher security and higher volume cryptographic operations.
hardware token	A hardware device (typically a smart card or USB device) that contains the user's digital ID(s) and requires a password or other authentication method to access those IDs for the purpose of signing.
HSM	See hardware security module.
ICA	See intermediate certificate authority (ICA).
IDABC	Interoperable Delivery of European eGovernment Services to public Administrations Businesses and Citizens individual digital ID: A digital ID issued to an individual to digitally sign as them self (e.g. John Smith) as opposed to an organization or other non-human entity.
individual digital ID	A digital ID issued to individuals so that they can identify themselves during a digital signature process (e.g. John Smith) as opposed to an organization or other non-human entity.
intermediate certificate authority (ICA)	A type of CA characterized by the fact that the ICA's certificate may itself be signed by a different ICA, all the way up to a 'self-signed' root certificate. Certificates in between the end entity and root certificates are sometimes called "intermediate certificates" (ICAs) and are issued by the CA or ICAs underneath the CA.
ISO	International Standards Organization.
ITC	Information Communication Technologies.

Table 8 Security Terms

long term validation	The validation of a digital signature after certificate(s) associated with the signature has expired.
LTV	See long term validation.
message digest	Before Acrobat or Adobe Reader can verify if a document the signed version of the document has changed or not (has integrity), it must first have a way to uniquely identify what was signed. To do this, it uses a message digest. A message digest is a number which is created algorithmically from a file and which uniquely represents that file. If the file changes, the message digest changes. Sometimes referred to as a checksum or hash, a message digest is simply a unique number created at signing time that identifies what was signed and is then embedded in the signature and the document for later verification.
MS	Member State (one of the EU countries)
MSCAPI	Windows Microsoft Crypto API (MSCAPI) is the API that the application uses to access cryptographic service providers such as PFX files and PKCS#11 files. MSCAPI is also used by the application anytime it uses a Windows security feature.
National Institute of Standards	A non regulatory agency of the United States Department of Commerce's Technology Administration. The institute's mission is to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve quality of life.
NIST	See National Institute of Standards.
OCSP	See online certificate status Protocol (OCSP).
online certificate status Protocol (OCSP)	OCSP defines a protocol for determining the revocation status of a digital certificate without requiring a CRL. Unlike CRL, OCSP obviates the need to frequently download updates to keep certification status lists current. Acrobat's OCSP revocation checker adheres to RFC 2560.
organizational digital ID	A digital ID issued to an organization or non-human entity (for example, the Adobe Public Relations Department). It can be used by an authorized employee / process to perform signing operations, at the desktop or server, on behalf of the company.
PAdES	See PDF Advanced Electronic Signatures.
PDF Advanced Electronic Signatures	A five part standard (ETSI TS 102 778) which describes how to use the digital signature features of the Portable Document Format (PDF) to meet EU signature requirements.
PKC	See public key certificate.
PKCS	*A* group of Public Key Cryptography Standards authored by RSA Security
PKCS#11	Public key Cryptography Standard #11: A Public key cryptography Standard published by RSA Laboratories defining an API, called Cryptoki, to devices which hold cryptographic information and perform cryptographic functions.
PKCS#11 device	External hardware such as a smart card reader or token. It is driven by a module (a software driver such as a .dll file on Windows).
PKCS#11 digital ID	An ID on a PKCS11# device. A device may contain one or more IDs.
PKCS#11 format	Cryptographic Token Interface Standard: An encryption format used by smart cards, tokens, and other PKCS#11-compatible devices. The ID is stored on the device rather than on the user's computer.
PKCS#11 module	The software module that drives a PKCS#11 device.
PKCS#11 token	See PKCS#11 device.

Table 8 Security Terms

PKCS#12	Public Key Cryptography Standard #12: Personal Information Exchange Syntax Standard that specifies a portable, password protected, and encrypted format for storing or transporting certificates. The certificates are stored in .pfx (Windows) and .p12 (Macintosh) files. Unlike other formats, the file may contain private keys.
PKCS#7	Public Key Cryptography Standard #7: A Public key cryptography standard published by RSA Laboratories that defines the syntax/format for a digital signature. This format extends PKCS#1 information to include timestamps, digital certificates and more. Files with .p7b and .p7c extensions are registered by the Windows OS. If you double click on a .p7c file it will be viewed by a Windows application. Replaced by CMS.
PKCS#9	Public Key Cryptography Standard #9 (of #15 produced by RSA) Selected Object Classes and Attribute Types. Defines attributes that PKCS#7 uses.
PKI	See public key infrastructure.
point of single contact	The point of single contact for a member state.
Policy Server	As of Acrobat 9, Adobe Policy Server is renamed to Adobe LiveCycle Rights Management Server
Private key	The secret key in a PKI system, used to validate incoming messages and sign outgoing ones. A Private Key is always paired with its Public Key during those key generations.
privileged context	A context in which you have the right to do something that’s normally restricted. Such a right (or privilege) could be granted by executing a method in a specific way (through the console or batch process), by some PDF property, or because the document was signed by someone you trust. For example, trusting a document certifier’s certificate for executing JavaScript creates a privileged context which enables the JavaScript to run where it otherwise would not.
PSC	See point of single contact.
public key	The publicly available key in a PKI system, used to encrypt messages bound for its owner and to validate signatures made by its owner. A Public Key is always paired with its Private Key during those key generations.
public key certificate	A file that contains the numeric public key portion of a public/private key pair along with the associated extensions and attributes that are used to define who the certificate is for, what its validity period is, and how the certificate can be used.
public key infrastructure	Term that includes all the CSPs, Certificates, and standards for encryption and digital signatures using public/private key pairs. Also, an arrangement that provides for trusted third party vetting of, and vouching for, user identities. It also allows binding of public keys to users. This is usually carried out by software at a central location together with other coordinated software at distributed locations. The public keys are typically in certificates.
QC	See qualified electronic certificate.
QEC	See qualified electronic certificate.
QES	See qualified electronic signatures.
QSCP	See qualified certificate service provider.
qualified certificate service provider	A CSP that has met the high assurance requirements spelled out in the EU Signature Directive as well as the implementing Member State's legislation, and provides credentials under a high assurance mechanism that includes secure signature creation devices.
qualified electronic certificate	A digital certificate from a QCSP that conforms to the RFC 3739 specification. It contains a qc statement that simply states that it is a qualified certificate. These types of certificates meet the requirements of the EU Signature Directive.

Table 8 Security Terms

qualified electronic signatures	Electronic signatures made by an secure signature-creation device with a QEC provided by a QCSP.
roaming ID	A roaming ID is a digital ID that is stored on a server. The private key always remains on the server, but the certificate and its public key can be downloaded at the subscriber's request to any location. Roaming IDs require an Internet connection and require the user to authenticate to the server to initiate the signature process. They eliminate the need to provide hardware tokens to users for private key storage.
root certificate	The top-most issuing certificate in a certificate chain; sometimes used as a trust anchor.
RSA	An encryption algorithm used to create a digital signature. The acronym derives from the creator's last names. In this case Rivest, Shamir and Adelman.
Secure Identity Across Borders Linked (STORK)	An EU project to better leverage cross-border trust and usage of eIDs.
secure signature creation device	A high assurance hardware device (smart card, USB token, etc) that stores the private key associated often with a QEC.
security restricted property or method	A property or method whose availability is restricted to certain events such as batch processing, console execution, or application startup. For example, in Acrobat 7.0, a security-restricted method (S) can only be executed through a menu event if one of the following is true: The JavaScript user preferences item "Enable menu items JavaScript execution privileges" is checked or the method is executed through a trusted function. The *JavaScript for Acrobat API Reference* identifies the items that have restrictions.
signature algorithm	The combined usage of a digest method (e.g. MD5, SHA1) and an encryption algorithm (e. g. DSA or RSA)
signature image	A type of electronic signature where the signer signs a document by physically applying their handwritten signature to a document using a dedicated signature pad or Tablet PC and plugin to Acrobat or Reader. These signatures may also digitally sign the document at the time of signing to protect integrity.
SSCD	See secure signature creation device.
STF 364	Special Task Force #364 Group in ETSI/ESI working on PDF AdES or PAdES.
STORK	Secure Identity Across Borders Linked: An EU project to better leverage cross-border trust and usage of eIDs.
SubCA	See subordinate certificate authority.
subordinate certificate authority	A type of CA characterized by the fact that the ICA's certificate may itself be signed by a different ICA (thus 'subordinate' to it), all the way up to a root certificate.
timestamp	A digitally signed timestamp whose signer vouches for the existence of the signed document or content at the time given as part of the digital signature. The time stamp data can be embedded in the digital signature using a trusted time server (instead of the time clock of the computer that is used to apply the digital signature). See also TSP.
TL	See trust list.
Trust (Trust service) Status List	As described in ETSI TS 102 231, a format and method for communicating in human and machine readable form all of the certificates trusted by each member state and their QCSPs.
trust anchor	A certificate in a certificate chain that is trusted for selected operations. It could be an intermediate certificate authority rather than a root; that is, it does not have to be the topmost certificate in the chain. Certificates that chain up to this certificate will also be trusted for the same operations. It is usually issued by a 3rd party certificate authority.
trust list	A list of trusted CA's and certificates.

Table 8 Security Terms

TS	See timestamp.
TSL	See Trust (Trust service) Status List.
TSP	A timestamp protocol which is described in RFC 3161.
X.509v3	In cryptography, X.509 is an ITU-T standard for a public key infrastructure (PKI) for single sign-on (SSO) and Privilege Management Infrastructure (PMI). X.509 specifies, amongst other things, standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm.
XAdES	See XML Advanced Electronic Signatures.
XML Advanced Electronic Signatures	An EU Advanced Electronic Signature Format relying on XML signatures, as described in ETSI TS 101 903.

Index

.

. 86, 87, 88
.apf 85
.cer 85
.p12 85
.p7b 85
.p7c 85
.pfx 85

6

64 bit Windows 53

A

A Custom Certificate Preference Example for Identrus Compliance 70
aAddressBook 73
AATL 85
AC 85
Accessing the Windows Certificate Store 35
Acrobat 3.0 and later 29, 31
Acrobat 5.0 and later 29, 32
Acrobat 6.0 and later 29, 32
Acrobat 7.0 and later 29, 32
Acrobat 9.0 and later 29, 32
Adding a contact 38
Adding Certificate Preferences 69
Adding or Removing Group Contacts 38
Adding recipients to a document with certificate security 41
aDefDirectory 64
AdES 85
aDirectory 73
Adobe LiveCycle Rights Management Server security 12
Adobe Policy Server 85
Adobe Profile Files 85
advanced electronic signatures 85
aHandler 80
AIA 85
AIIM 85
ALCRMS 85
ALCRMS Server Configuration 50
All contents
14
All contents except metadata
14
Allowing Self Sign Digital IDs 58
Always use this digital ID 40, 44
Applying a Certificate Security Policy 46
Applying a Certificate Security to a Group 46
Applying a Security Policy to a Document 23
Applying ALCRMS Security 51
approval signature 85
aPrivKey 72
Ask me which digital ID to use next time 40, 44
ASPKI 86
ASPKI Service Provider Interfaces 73
Associating a Certificate Preference with a Chain Scope 69
Associating Batch Processing with a Security Method 17
Association for Information and Image Management 86
Audit alert for ALCRMS security 51
authority information access 86
aVerify 72

B

bAllowAPSConfig 55
bAllowConnectViaHTTP 81
bAllowInvisibleSig 55
bAllowPasswordSaving 54, 78
basic constraint 86
bAskBeforeInstalling 57
bCertStoreImportEnable 66
bCustomPrefsCreated 64
Because You're Lost? 9
Because Your Technical 6
bFDFExportSave 76
bFDFExportSign 76
bFDFRequestExcludeCert 76
bFDFRequestSave 76
bFIPSMode 79
bIsFavorite 80
bLoadSettingsFromURL 57, 67
bPrivKey 54
bReasons 54
bReqRevCheck 54
bSavePassword 82
bSelfSignCertGen 58
bShowAllChains 77
bSuppressStatusDialog 54
bValidateOnOpen 54
bVerify 54
bVerifyUseAlways 54, 72
bWebBuyFDF 76
bWinCacheSessionHandles 55
by ASPKI. 71

C

CA 86
cAcceptablePolicyOIDsc1 71
cAcceptablePolicyOIDsc1cValue 71
cAccounts 60
cACs 63
CAdES 86
CAPI 86
cAuthMechanisms 75
cCertIssuerInfo 58
cCredentials 61

cDialogs
  xSelHandler 73
cDigitalIDFiles 63
cDirectoryProvider 75
CDS 86
CDS digital ID 86
CDS digital ID certificate 86
cEmbeddedFilePath 65
CEN 86
certificate 86
certificate authority (CA) 86
Certificate Management 65
certificate revocation list (CRL) 86
Certificate Viewer details tab 68
certification signature 86
certified document 86
Certified Document Services (CDS) 86
certify or certifying 86
cExportPath 65
cExtendedCertInfo 58
Changes Allowed 16, 30, 42, 43, 45
Changing and Viewing Security Settings 17
Changing Document Collection Passwords 34
Changing the Security Method Type 20
Choosing a certificate for encryption 37
Choosing a digital ID for certificate security 40
Choosing a Security Method Type 12
Choosing an Algorithm 14
Choosing What to Encrypt 14
cImportPath 65
Click thru signature 86
cModules 59
CMS 86
Compatibility 29
Configure the Select Document Components to Encrypt 30, 32, 39, 43
Configuring ALCRMS Settings Manually 49
Configuring Certificate Handling for Certificate Security 80
Configuring LiveCycle Right Management Servers 81
Configuring Password Caching 78
Configuring Servers 48
Contacts
  Editing a group 38
Content Security 79
cOpenCategories 77
Copying a Security Policy 24
cP11Credentials 59
cP11Path 59
cProfilePath 65
Creating a Group 37
Creating a Reusable Certificate Security Policy 38
Creating a Reusable Password Security Policy 28
Creating an ALCRMS Security Policy 50
Creating Certificate Security for the Current Document 42
Creating Certificate Security Settings 38
Creating Password Security for One-Time Use 31
Creating Password Security Settings 28
Creating Security Policies with Policy Manager 22
cRecentServerURLs 60
cRevocationChecker 74
CRL 87
CRL cache 74
cryptographic service provider 87
cSAML_Assertion 61
CSP 87
Custom certificate preference key 69
Custom Certificate Preferences 67
Custom Security Handler Preferences 71

D

Deleting a Security Policy 25
digest method 87
digital ID 87
Digital ID Import and Export Paths 64
Digital ID Management 58
digital signature 87
digital signature algorithm 87
digitally sign 87
Displaying All Chains in the Certificate Viewer 77
document integrity 87
Document open password 28
Document Property dialog 20
Document security settings
  ALCRMS security 19
  Certificate security 18
  Password security 18
Documentation related to security 9
DSA 87

E

EC 87
Editing a Security Policy 24
Editing Security Method Settings 21
EE 87
EESSI 87
eID 87
electronic ID 87
electronic signature 87
embedded JavaScript 87
embedded validation response 88
Enable text access for screen reader devices for the visually impaired 17, 31, 42, 44, 46
Encryption Algorithms 8
Encryption configuration panel 14
Encryption Workflow 13
end entity certificate (EE) 88
Envelopes 26
ETSI 88
ETSI/ESI 88
EU 88
EU Experts Group 88
EU Signature Directive 88

F

FDF Import and Export 76
FDF preferences 76

FIPS 88
FIPS 140 88

H

hardware security module 88
hardware token 88
HSM 88

I

iAuthMechanisms 75
ICA 88
iCheckEvery 57
IDABC 88
Identifying Certificates with a Hash of the Public Key 67
iDirectoryProvider 75
iEnd 69
Importing ALCRMS Settings from an FDF file 49
Importing ALCRMS Settings with a Security Settings Import 49
Importing Security Settings 56
iMSStoreTrusted 66
individual digital ID 88
intermediate certificate authority (ICA) 88
iResourceID 57
iRevocationChecker 74
iSHS 77
ISO 88
iStart 69
iSVS 77
ITC 88

K

Kerberos Authentication Mechanism 62

L

Lockdown keys 55
long term validation 89
LTV 89

M

Making a Security Policy Favorite 25
Managing your ALCRMS Account 50
message digest 89
MS 89
MSCAPI 89

N

National Institute of Standards 89
NIST 89

O

OCSP 89
online certificate status Protocol (OCSP) 89
Only file attachments
14
Opening a Certificate-Protected Document 47
Opening a Password-Protected Document 33
organizational digital ID 89
Organizational policies 22

P

PAdES 89
Password caching
    Disabled UI item 78
Password prompt 33
Password Recovery 34
Password security 12
PDF Advanced Electronic Signatures 89
Permissions Options 16
Permissions panel 15
Permissions password 28
Permissions Workflow 15
PKC 89
PKCS 89
PKCS#11 89
PKCS#11 Configuration 59
PKCS#11 device 89
PKCS#11 digital ID 89
PKCS#11 format 89
PKCS#11 module 89
PKCS#11 token 89
PKCS#12 90
PKCS#7 90
PKCS#9 90
PKI 90
Policy favorites preferences 80
Policy security method selection 23
Policy Server 90
Preference relevance keys 70
Preventing End-User Modification 53
Printing Allowed 16, 30, 41, 43, 45
Private key 90
privileged context 90
PSC 90
public key 90
public key certificate 90
Public key certificate security 12
public key infrastructure 90

Q

QC 90
QEC 90
QES 90
QSCP 90
qualified certificate service provider 90
qualified electronic certificate 90

R

Refreshing the Security Policy List 25
Registry preferences
    cPPKHandler 63
    Digital ID 58

- Digital ID path 65
- FDF import and export 76
- FIPS mode 79
- Identrus 71
- Kerberos 62
- LiveCycle Right Management Server 81
- LiveCycle Right Management Server list 82
- MSCAPI provider 66
- Password caching 78
- PKCS#11 59
- Roaming ID account 60
- Roaming ID server 60
- Roaming ID user account (set by the server) 61
- Scoping certificate 69
- Security envelope 82
- Security handler 72
- Security setting import 57
- Security Settings Console 77
- Self-signed digital ID 58
- Signature validation 77
- SPI for authentication mechanisms 75
- SPI for directory providers 75
- SPI for revocation checker providers 74
- subject to feature lockdown 54
- Trusted Identity Manager 67

Removing a contact 38
Removing Document Security 21
Removing Password Security 33
Requiring FIPS-Compliant Encryption Algorithms 80
Resource roadmap 9
Revocation Checker Providers 73
Revoking a Document 52
roaming ID 91
Roaming ID Authentication Mechanisms 75
Roaming ID Configuration 59
Roaming ID Provider Persistent Storage 60
Roaming ID Server-Set Preferences 60
root certificate 91
RSA 91

S

Searching for group contacts 47
Secure Identity Across Borders Linked (STORK) 91
secure signature creation device 91
Security envelope 26
Security Envelopes 82
Security Method Basics 11
Security method pros and cons 13
Security method selection 12
Security method selection from Policy Manager 22
Security method workflow 11
Security Methods and Encryption 13
Security Methods and Permissions 15
Security methods for batch processing 17
Security Policies 13
- Reusable Security Settings 21

Security policy
- Favorites list 25
- General settings 23, 29, 39

security restricted property or method 91
Security settings
- Export dialog 56

Security Settings Console 77
Security settings icon 20
Security settings require “save” alert 33
Security Terms 85
Select Document Components to Encrypt panel 14
Selecting a Certificate to Use for Encryption 36
Set the compatibility level 31
Setting Basic Client and Workflow Preferences 53
Setting Digital ID Defaults 63
Setting up Security Policy Favorites 80
Setting up the Certificate Security Environment 35
sId 80
signature algorithm 91
signature image 91
Signature Validation Directory Providers 74
Specifying extended certificate information 58
Specifying preference relevance 70
SSCD 91
sServiceName 62
Standards support 83
STF 364 91
STORK 91
SubCA 91
Synchronizing a Document for Offline Use 52

T

tAddressBook 67
tCredProvider 63, 64
tFriendlyName 61
timestamp 91
TL 91
tLastEnvelope 82
tLastSearchPath 82
tLastServerURL 81
tLoadSettingsURL 57
tLockboxId 82
tName 80, 82
Trust (Trust service) Status List 91
trust anchor 91
trust list 91
Trusted Identity List Configuration 66
tSAML_Assertion_Expiration 61
tSAML_Assertion_Source 61
tSAML_Name_Format 62
tSAML_Name_Qualifier 62
tSAML_Name_Value 61
tSASL_Mechanism 62
tSASL_UserName 62
tServer 82
tServerName 60, 81
tServerURL 61, 81
TSL 92
TSP 92
tURL 60

Turning on FIPS Mode 78

U

Use this digital ID until I close the application 40, 44
User policies 22

V

Validity period expired alert 51
View a Document’s Audit History 52
Viewing a Security Policy 24
Viewing Document Encryption and Permission Settings 18
Viewing Document Restrictions 19
Viewing Security Settings in a Browser 20

W

What’s new for 9.1? 6
Windows Integration 65
Windows integration 36
Working with Documents and ALCRMS Policies 50
Working with Groups of Contacts 37

X

X.509v3 92
XAdES 92
xCategory 77
xCertSHA1 64
xdata 57
xDefEnrollmentURL 63