



PDF Creation Date:

November 17, 2008

Digital Signature User Guide

for Acrobat 9.0 and Adobe Reader 9.0

Acrobat® and Adobe® Reader®

Version 9.0

© 2008 Adobe Systems Incorporated. All rights reserved.

Digital Signature User Guide for Adobe® Acrobat 9.0 and Adobe® Reader 9.0 on Windows® and Macintosh®.

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

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, Acrobat, Reader, and the Adobe logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Windows®, Windows NT®, and Windows XP® are registered trademarks of Microsoft® Corporation registered in the United States and/or other countries. Mac® and Macintosh® are registered trademarks of Apple Computer®, Inc. in the United States and other countries. All other trademarks are the property of their respective owners.

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	Getting Started	8
1.1	What's in this Guide?	8
1.2	Who Should Read This Guide?	8
1.3	How Should You Use This Guide?	9
1.4	Roadmap to Other Security Documentation	9
2	Getting and Using Your Digital ID.....	11
2.1	Digital ID Basics.....	11
2.1.1	What is a Digital ID?	11
2.1.2	Digital ID Storage Mechanisms	12
2.1.3	Registering a Digital ID for Use in Acrobat	13
2.1.4	Digital ID Management and the Security Settings Console.....	14
2.1.5	Setting Identity Information.....	14
2.2	Generic ID Operations	15
2.2.1	Specifying Digital ID Usage.....	15
2.2.2	Sharing (Exporting) a Digital ID Certificate	16
2.2.3	Viewing All of Your Digital IDs	16
2.2.4	Customizing a Digital ID Name	17
2.2.5	Viewing Digital ID Certificates in the Certificate Viewer	17
2.3	Managing PKCS#12 Digital ID Files	19
2.3.1	Logging in to a Digital ID File.....	19
2.3.2	Adding a Digital ID from a PKCS#12 File	19
2.3.3	Adding and Removing Digital ID Files from the File List.....	20
2.3.4	Changing an ID File's Password	20
2.3.5	Changing a PKCS#12 File's Password Timeout.....	21
2.3.6	Logging in to PKCS#12 Files	22
2.3.7	Creating a Self-Signed Digital ID.....	22
2.3.8	Deleting a PKCS#12 Digital ID.....	25
2.4	Managing Windows Digital IDs	26
2.4.1	Finding a Digital ID in a Windows Certificate Store File	26
2.4.2	Deleting a Digital ID from the Windows Certificate Store	26
2.5	Your server may require additional or different authentication steps. Follow directions that appear in the dialogs.Managing IDs Stored on Hardware Devices	27
2.5.1	Adding an ID that Resides on External Hardware.....	27
2.5.2	Changing Passwords	28
2.5.3	Logging in to a Device.....	28
3	Managing Certificate Trust and Trusted Identities	30
3.1	What is Trust?	30
3.2	What is a Trusted Identity?	30
3.3	Adding Someone to Your Trusted Identity List.....	32
3.3.1	Requesting a Digital ID via Email	33
3.3.2	Importing a Certificate From a File.....	33
3.3.3	Searching for Digital ID Certificates	34
3.4	Certificate Trust Settings.....	35

3.4.1 Using Certificates for Certificate Security (Encryption).....	38
3.5 Using Directory Servers to Add Trusted Identities	38
3.5.1 Manually Configuring a Directory Server.....	39
3.5.2 Editing Directory Servers Details	40
3.5.3 Deleting a Directory Server	41
3.5.4 Specifying a Default Directory Server	41
3.5.5 Importing and Exporting Directory Server Settings	41
3.6 Managing Contacts.....	42
3.6.1 Viewing and Editing Contact Details.....	42
3.6.2 Emailing Certificate or Contact Data	43
3.6.3 Saving Certificate or Contact Details to a File	43
3.6.4 Associating a Certificate with a Contact.....	43
3.6.5 Changing a Trusted Identity's Certificate Association	44
3.6.6 Deleting Contacts and Certificates.....	44
4 Authoring Signable Documents.....	46
4.1 Best Practices for Signed Documents that will Change	46
4.2 Setting up the Signing Environment.....	46
4.2.1 Setting Signing Preferences	47
4.2.1.1 Requiring Preview Mode.....	47
4.2.1.2 Changing the Default Signing Method	48
4.2.1.3 Embedding Signature Revocation Status	49
4.2.1.4 Allowing Signing Reason	50
4.2.1.5 Showing Location and Contact Details	50
4.2.1.6 Enabling Document Warning Review	50
4.2.1.7 Requiring Document Warning Review Prior to Signing	51
4.2.1.8 Enabling a Warnings Comment or Legal Attestation	51
4.2.2 Customizing Signature Appearances.....	52
4.2.2.1 Creating a Custom Signature	52
4.2.2.2 Creating a Custom Watermark or Background.....	52
4.2.2.3 Creating a Custom Signature Appearance.....	53
4.2.2.4 Editing or Deleting a Signature Appearance	54
4.2.3 Using Timestamps During Signing	55
4.3 Working with Signature Fields	56
4.3.1 Creating a Blank Signature Field.....	57
4.3.2 Specifying General Field Properties.....	58
4.3.3 Customizing Field Appearances	59
4.3.4 Changing the Default Field Appearance	60
4.3.5 Cut, Copy, and Paste Signature Fields	60
4.3.6 Arranging Signature Fields	60
4.3.7 Creating Multiple Copies of a Signature Field	61
4.4 Authoring Signable Forms	62
4.4.1 Authoring a Document with Multiple Fields.....	62
4.4.2 Locking Fields Automatically After Signing.....	62
4.4.3 Making a Field a Required Part of a Workflow	63
4.4.4 Specifying a Post-Signing Action.....	64
4.4.5 Unlocking a Field Locked by a Signature.....	66
5 Controlling Signing with Seed Values	67
5.1 Seed Value Basics.....	67
5.1.1 Changes Across Releases	68

5.1.2 Supported Seed Values	69
5.1.3 Enabling JavaScript to Set Seed Values	70
5.2 Forcing a Certification Signature	71
5.3 Giving Signers the Option to Lock a Document	73
5.4 Forcing Signers to Use a Specific Signature Appearance	74
5.5 Adding Custom Signing Reasons	75
5.6 Specifying Timestamps for Signing	76
5.7 Specifying Alternate Signature Handlers and Formats	77
5.8 Specifying a Signature Hash Algorithm	79
5.9 Embedding Revocation Information in a Signature	79
5.10 Specifying Certificate Properties for Signing	80
5.10.1 Specifying Signing Certificates Origin	82
5.10.2 Specifying Certificates by Key Usage	83
5.10.3 Specifying Certificates by Policy	84
5.10.4 Specifying a URL When a Valid Certificate is not Found	85
5.10.5 Restricting Signing to a Roaming ID	86
5.11 Custom Workflows and Beyond	86
6 Signing Documents	89
6.1 Signing Basics	89
6.1.1 Before You Sign	89
6.1.2 Signature Types	89
6.1.3 Signing User Interface	90
6.2 Signing With a Certification Signature	90
6.2.1 Certification Workflow for Documents with Multiple Signers	92
6.2.2 Setting up a Document for Certification	93
6.2.3 You can customize the way a certified document behaves for signers by giving form fields additional features with seed values. For example, you can preconfigure custom signing reasons or limit signing to only those with certificates with predefined characteristics. Certifying a Document .	93
6.2.4 Certifying a Dynamic Form	95
6.2.5 Why Can't I Certify?	96
6.3 Signing with an Approval Signature	96
6.3.1 Signing Documents in Acrobat	96
6.3.2 Signing in a Browser	98
6.3.3 Clearing One or More Signatures	99
7 Validating Signatures	100
7.1 Signature Validity Basics	100
7.1.1 What Makes a Signature Valid?	100
7.1.1.1 Authenticity Verification	101
7.1.1.2 Document Integrity Verification	101
7.2 Setting up Your Environment for Signature Validation	102
7.2.1 Validating Signatures Automatically	102
7.2.2 Setting Digital Signature Validation Preferences	103
7.2.3 Using Root Certificates in the Windows Certificate Store	104
7.2.4 Validating Signatures with Timestamps and Certificate Policies	105
7.3 Validating Signatures Manually	106
7.3.1 Validating Signatures with Adobe Reader	106
7.3.2 Validating a Single Signature in Acrobat	106
7.3.3 Validating All Signatures in Acrobat	107

7.3.4	Validating an Problematic Signature (trusting a signer on-the-fly)	108
7.3.5	Validating Signatures for other Document Versions	110
7.3.6	Validating Signature Timestamps	110
7.3.6.1	When Timestamps Can't be Verified. . .	112
7.4	Status Icons and Their Meaning	113
7.4.1	Signature Status Definitions	113
7.4.2	Document Status Definitions	113
7.5	Troubleshooting a Signature or Document Status	115
7.5.1	Troubleshooting an Identity Problem	115
7.5.1.1	Troubleshooting Digital ID Certificates	116
7.5.1.2	Displaying the Signer's Certificate	117
7.5.1.3	Verifying the Identity of Self-Signed Certificates	118
7.5.1.4	Checking Certificate Revocation Status	119
7.5.1.5	Exporting a Certificate Other than Yours to a File	120
7.5.2	Troubleshooting a Document Integrity Problem	120
7.5.2.1	LiveCycle Dynamic Forms and the Warning Triangle	121
7.5.2.2	Viewing and Comparing Changes and Versions	122
7.5.2.3	Viewing a List of Post-Signing Modifications	122
7.5.2.4	Comparing a Signed Version to the Current Version	123
7.6	Document Behavior After Signing	124
7.6.1	JavaScript and Dynamic Content Won't Run	125
7.6.2	Certifying a Document is Prevented	125
7.6.3	Form Field Fill in, Signing, and/or Other Actions Don't Work	125
8	Document Integrity and Preview Mode	126
8.1	Preview Mode and Signing Workflows	126
8.2	Preview Mode and Validation (View Signed Version)	127
8.3	PDF Signature Reports	127
8.4	Signature Report Error Codes	129
9	External Content and Document Security	132
9.1	Enhanced Security	132
9.1.1	Enabling Enhanced Security	133
9.1.2	Changes in FDF Behavior	134
9.1.3	Interaction with Trust Manager	136
9.1.4	Make Privileged Folder Locations Recursive	136
9.2	Controlling Multimedia	136
9.2.1	Configuring Multimedia Trust Preferences	137
9.2.2	Controlling Multimedia in Certified Documents	138
9.3	Setting JavaScript Options	139
9.3.1	High Privilege JavaScript Defined	139
9.3.2	Javascript and Certified Documents	139
9.4	Adobe Trusted Identity Updates	140
9.5	Working with Attachments	141
9.5.1	Default Behavior: Black and White Lists	141
9.5.2	Adding Files to the Black and White Lists	144
9.5.3	Resetting the Black and White Lists	144
9.5.4	Allowing Attachments to Launch Applications	145
9.6	Controlling Access to Referenced Files and XObjects	145
9.7	Internet URL Access	146
9.7.1	Turning Internet Access Off and On	146

9.7.2 Allowing and Blocking Specific Web Sites	147
10 Migrating and Sharing Security Settings	149
10.1 Security Setting Import and Export	149
10.1.1 Exporting Security Settings to a File	149
10.1.2 Importing Security Settings from a File	150
10.1.3 Importing Security Settings from a Server	152
10.2 Sharing Settings & Certificates with FDF	152
10.2.1 FDF Files and Security	154
10.2.2 Exporting Application Settings with FDF Files	155
10.2.2.1 Distributing a Trust Anchor or Trust Root	155
10.2.2.2 Setting the Certificate Trust Level	158
10.2.2.3 Exporting Your Certificate	158
10.2.2.4 Emailing Your Certificate	159
10.2.2.5 Saving Your Digital ID Certificate to a File	160
10.2.2.6 Requesting a Certificate via Email	161
10.2.2.7 Emailing Server Details	162
10.2.2.8 Exporting Server Details	163
10.2.3 Importing Application Settings with FDF Files	164
10.2.3.1 Responding to an Email Request for a Digital ID	164
10.2.3.2 Importing Someone's Certificate	166
10.2.3.3 Importing Multiple Certificates	167
10.2.3.4 Importing Timestamp Server Settings	169
10.2.3.5 Importing Directory Server Settings	171
10.2.3.6 Importing Adobe LiveCycle Rights Management Server Settings	172
10.2.3.7 Importing Roaming ID Account Settings	173
10.2.3.8 Importing a Trust Anchor and Setting Trust	175
11 Glossary of Security Terms	178
12 Index	182

1 Getting Started

1.1 What's in this Guide?

This guide describes the digital signature features of the Acrobat 9.0 family of products:

- Working with digital IDs that are used for signing and certificate security workflows:
 - [Chapter 2, "Getting and Using Your Digital ID"](#)
 - [Chapter 3, "Managing Certificate Trust and Trusted Identities"](#)
- Digital signatures:
 - [Chapter 4, "Authoring Signable Documents"](#)
 - [Chapter 6, "Signing Documents"](#)
 - [Chapter 7, "Validating Signatures"](#)
- Migrating and sharing security settings:
 - ["Security Setting Import and Export" on page 149](#)
 - ["Sharing Settings & Certificates with FDF" on page 152](#)
- Securing the application environment:
 - ["Controlling Multimedia" on page 136](#)
 - ["Setting JavaScript Options" on page 139](#)
 - ["Working with Attachments" on page 141](#)
 - ["Controlling Access to Referenced Files and XObjects" on page 145](#) (only available in 7.0.5 and later)
 - ["Internet URL Access" on page 146](#)

1.2 Who Should Read This Guide?

End users: This document describes how to configure and use the application user interface for signing and signature validation, register a digital ID for use in Acrobat, and manage other people's public key certificates within your system.

Administrators: This document describes how to configure and use the application user interface. Because system administrators may be responsible for deploying and supporting the Adobe Acrobat family of products (including Adobe Reader) in digital signature workflows, leverage this guide to help your clients use the product correctly and effectively. This guide should be used in conjunction with the *Acrobat Security Administration Guide*.

1.3 How Should You Use This Guide?

If you are setting up a signature workflow for the first time, do not have a digital ID, or have not established some sort of trust for other signer's whose signature you need to validate, read [Chapter 2, "Getting and Using Your Digital ID"](#) and [Chapter 3, "Managing Certificate Trust and Trusted Identities"](#). In enterprise settings, the administrator may issue you an ID (or provide instructions on getting one) and may also set up your application so that it can verify (trust) signatures.

- **Authoring:** If you are a document author or manage document templates that contain signature fields, read [Chapter 4, "Authoring Signable Documents"](#).
- **Signing:** If you will be signing documents, configure your application and learn about the signature types and signing process as described in [Chapter 6, "Signing Documents"](#).
- **Validating Signatures:** If you will be validating signatures in signed documents, configure your application and learn about what controls signature status as described in [Chapter 7, "Validating Signatures"](#). If you have not previously configured a trust anchor, you may also want to read [Chapter 3, "Managing Certificate Trust and Trusted Identities"](#).
- **Configuring machines:** If you are concerned about securing the application environment and controlling document and application access to external content such as the Internet and attachments, see [Chapter 9, "External Content and Document Security"](#).
- **Sharing application settings:** If you need to share your certificate or server settings with someone, see [Chapter 10, "Security Setting Import and Export"](#).

1.4 Roadmap to Other Security Documentation

In many enterprise environments, there is no clear distinction between audience types. Some end users are "power users" and don't shy away from modifying the registry and tweaking applications in administrator-like ways. Some system administrators are highly technical and perform developer-like tasks such as PERL programming and JavaScript scripting. For this reason, it is up to the reader to determine what documents listed in [Table 1](#) are pertinent to their tasks. However, this document uses the following definitions:

- **User or end user:** End users usually have their application installed and preconfigured by an administrator. They only interact with the graphical user interface and do not modify the registry. Some end users, such as document authors, may use simple JavaScripts to set seed values on documents.
- **Administrator:** System administrators install and configure end user machines. More often than not, they use the installer wizard to configure the product installer prior to deploying applications across the enterprise. Because the end user experience can be controlled by the registry, administrators must be familiar with both the application's user interface and capabilities as well as the options for registry configuration.
- **Developer:** Developers typically try to find programmatic ways to generate or process PDF documents. They read specifications and API documents to figure out how to solve real-world enterprise problems without requiring manual human intervention. Communication with servers is often a requirement. Because enterprise solutions often involve understanding application behavior, developers sometimes need to review administration guides to learn how to deploy plugins or handlers and to learn how to configure the application to use those components. Many of the application's registry settings can be accessed and manipulated via JavaScript.

Note: The most recent document versions may be found online at:

- http://www.adobe.com/go/acrobat_developer
- http://www.adobe.com/go/acrobat_security

Table 1 Documentation related to Acrobat security

Document	Audience	For information about
<i>Acrobat SDK Documentation Roadmap</i>	Developers	A guide to the documentation in the Adobe Acrobat SDK.
<i>Acrobat and PDF Library API Reference</i>	Developers	A description of the APIs for Acrobat and Adobe Reader® plug-ins, as well as for PDF Library applications.
<i>JavaScript for Acrobat API Reference</i>	Developers	A listing of the Acrobat JavaScript APIs.
<i>Developing Acrobat Applications with JavaScript</i>	Developers	Additional detail about the Acrobat JavaScript APIs.
<i>PDF Reference 1.7</i>	Developers	A detailed description of the PDF language.
<i>FDF Data Exchange Specification</i>	Developers	A object-level FDF file description. The files can be generated programmatically and used to share security-related data.
<i>PDF Signature Build Dictionary Specification</i>	Developers	Build properties for the PDF Reference's signature dictionary which provides interoperability details for 3rd party handlers.
<i>Digital Signature Appearances</i>	Developers & administrators	Guidelines for creating signatures programmatically.
<i>Guidelines for Developing CSPs for Acrobat on Windows</i>	Developers & administrators	Guidelines for developing a Cryptographic Service Provider for use with Acrobat® on the Windows® platform.
<i>Acrobat Security Administration Guide</i>	Administrators	Application deployment and configuration in enterprise settings.
<i>Digital Signature User Guide for Adobe Acrobat and Adobe Reader</i>	Administrators & end users	Application usage and configuration via the user interface.
<i>Document Security User Guide Adobe Acrobat and Adobe Reader</i>	Administrators & end users	Application usage and configuration via the user interface.
<i>Security Setting User Guide</i>	Administrators & end users	Describes how to export and import security settings and certificate data with .acrobatsecurity and FDF files.
<i>Enhanced Security in Adobe Acrobat 9 and Adobe Reader 9</i>	Administrators & end users	X-domain configuration specifically and other aspects of the enhanced security feature generally.
<i>Digital Signatures in the PDF Language</i>	Anyone needing an overview	A generic description of how signature work in PDF.
<i>Digital Signatures in Acrobat</i>	Anyone needing an overview	A description of how signatures are implemented in Acrobat.
<i>Acrobat 9 Digital Signatures Changes and Improvements</i>	Anyone needing an overview	A description of the changes in digital signatures since Acrobat 8.

A digital ID is like a driver's license or passport or other "certified by some entity" paper identification. It proves your identity to people and institutions that you communicate with electronically. These IDs are a critical component of digital signatures and certificate security. In signing and certificate security workflows, you will be asked to select a digital ID. Selecting an ID is simply a matter of picking one from a list of your previously installed digital IDs. If you do not have a digital ID, you will be prompted to find or create one.

For more information, refer to the following:

- ["Digital ID Basics" on page 11](#)
- ["Generic ID Operations" on page 15](#)
- ["Managing PKCS#12 Digital ID Files" on page 19](#)
- ["Managing Windows Digital IDs" on page 26](#)
- ["Your server may require additional or different authentication steps. Follow directions that appear in the dialogs.Managing IDs Stored on Hardware Devices" on page 27](#)

2.1 Digital ID Basics

2.1.1 What is a Digital ID?

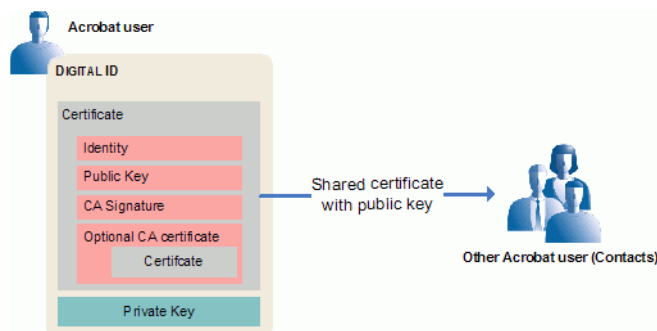
A digital ID consists of two main parts: a certificate and a private key. A certificate consists of your identity information (name, date, serial number, etc.) and a public key that are bound together and signed by a trusted or untrusted certificate authority. The certificate sometimes includes a reference to the certificate issuer's certificate, thereby creating what is known as a "certificate chain."

Digital IDs operate by using a key pair: data encrypted with one key can only be decrypted by the other corresponding key. When you sign PDF documents, you use the private key to apply your digital signature. You distribute the certificate that contains your public key to those who need to validate your signature or encrypt information for you. Only your private key can unlock information that was encrypted using your public key, so be sure to store your digital ID in a safe place.

You must have a digital ID to sign, certify, and apply certificate encryption to PDFs. You can get a digital ID from a third-party provider, or you can create a self-signed digital ID. Self-signed digital IDs may be adequate for many situations. However, to prove your identity in most business transactions, you may need a digital ID from a trusted third-party provider, called a certificate authority. Because the certificate authority is responsible for verifying your identity to others, choose one that is trusted by major companies doing business on the Internet.

You can have multiple digital IDs for different purposes. For example, you may sign documents in different roles or using different certification methods. Digital IDs are usually password protected and can be stored on your computer in password protected file, on a smart card or hardware token, in the Windows certificate store, or on a signing server (for roaming IDs). Acrobat applications can access digital IDs from any of these locations.

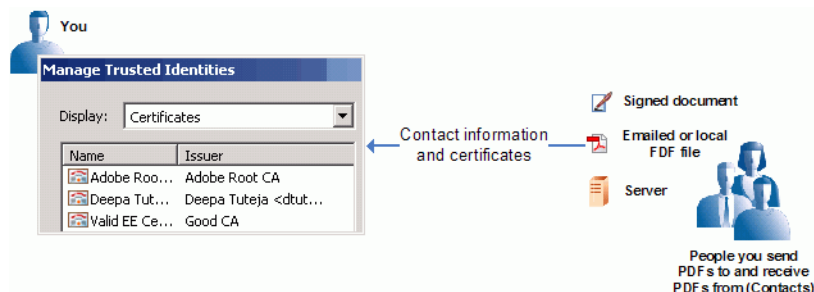
Figure 1 Digital ID: Components



Users exchange their digital ID's certificate so that they can validate signatures and encrypt documents for each other. Shared certificates can be physically sent in a file or made available over a network. The private key is never shared and is used to decrypt documents. There are several ways to share certificates:

- **Physical sharing:** Certificates can be physically shared in a file sent via email or located in a shared directory. They can be imported, exported, and otherwise managed with the Trusted Identity Manager. For details, see [Chapter 3, "Managing Certificate Trust and Trusted Identities"](#).
- **Network sharing:** Certificates can be stored on a central server. The Trusted Identity Manager can be used to search for certificates on LDAP directory servers. Adobe applications provide tools for configuring and managing directory servers. For details, see ["Using Directory Servers to Add Trusted Identities" on page 38](#).

Figure 2 Trusted identities



2.1.2 Digital ID Storage Mechanisms

A digital ID's certificate and private key need to be stored in a secure location. There are several file types and file locations where these items could be stored ([Table 2](#)). The digital ID data in these files is provided to the application via digital ID service providers (sometimes called Cryptographic Service Providers or CSPs). A service provider is simply a storage mechanism and code that makes the data available to the application.

In most cases, the digital ID is stored on a local or networked file. Common locations include the Windows Certificate Store which is accessible by Adobe applications and other Windows applications and the Acrobat store which is used only by the Acrobat family of products. Some IDs may exist only on external hardware such as a smart card connected to the computer.

The Acrobat family of products can access a digital ID from the following storage mechanisms:

- **Windows Certificate Store:** A local store (file location) provided by Windows that can import and export various file formats and that can be used by both Windows programs and Acrobat products.
- **PKCS#12 files:** A common file format residing on your hard drive that is used on both Windows and Macintosh.

Tip: PKCS refers to a group of Public Key Cryptography Standards authored by RSA Security

- **PKCS#11 devices:** External devices such as a USB token or smart card that store digital ID data.
- **Roaming ID servers:** A network server. The private key is known only to the server. The server sends the certificate and its public key to users on demand. Users can import and export the certificate and its public key from Acrobat, but they never have install the private key on a local machine.

Table 2 Digital ID-related file types

Type	Description	5.x	6.x	7.x	8.x	9.x
.acrobat security	An XML format encapsulated in a PDF which stores security settings for import and export. Contains: Digital ID (public and private keys)					Export Import
PKCS#12: .pfx (Win), .p12 (Mac)	Personal Information Exchange Syntax Standard: Specifies a portable, password protected, and encrypted format for storing or transporting certificates. Contains: Digital ID (public and private keys)		Export Import	Export Import	Export Import	Export Import
.fdf	An Adobe file data exchange format used for importing and exporting settings and certificates (usually PKCS#12 files).	Export Import	Export Import	Export Import	Export Import	Export Import
PKCS#7: .p7b, .p7c	Certificate Message Syntax (CMS): Files with .p7b and .p7c extensions are registered by the Windows OS. Acrobat products can import and export these files. Contains: Certificate and public key only		Export Import	Export Import	Export Import	Export Import
.cer	Certificate format: A Microsoft format for digital IDs usually stored in the Windows Certificate Store. Contains: Certificate and public key only		Export Import	Export Import	Export Import	Export Import
.apf	Adobe Profile Files (Legacy): Not used after Acrobat 5. Files can be upgraded by double clicking them. Contains: Digital ID (public and private keys)	Import Export	Import	Import	Import	n/a

2.1.3 Registering a Digital ID for Use in Acrobat

Digital IDs help you sign, certify, and apply certificate security to documents. There are two ways to register a digital ID:

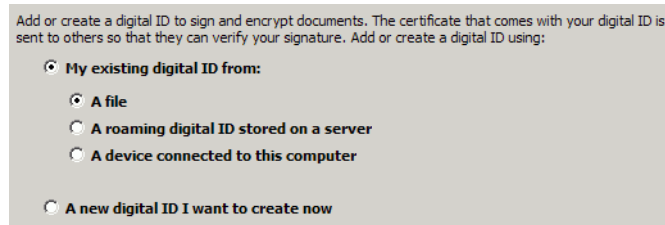
- **In advance:** You can set up the ID ahead of time for later use. To do so, choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**, selecting **Digital IDs** in the left-hand tree, and then choosing **Add ID**.
- **On the fly:** You can find or add IDs in signature and certificate security workflows. For example, when the Sign Document dialog appears, choose **New ID** from the **Sign As** drop down list.

For more information, refer to the following:

- [Adding a Digital ID from a PKCS#12 File](#)

- [Finding a Digital ID in a Windows Certificate Store File](#)
- [Adding an ID that Resides on External Hardware](#)
-

Figure 3 Add Digital ID dialog



.apf Digital IDs no longer supported

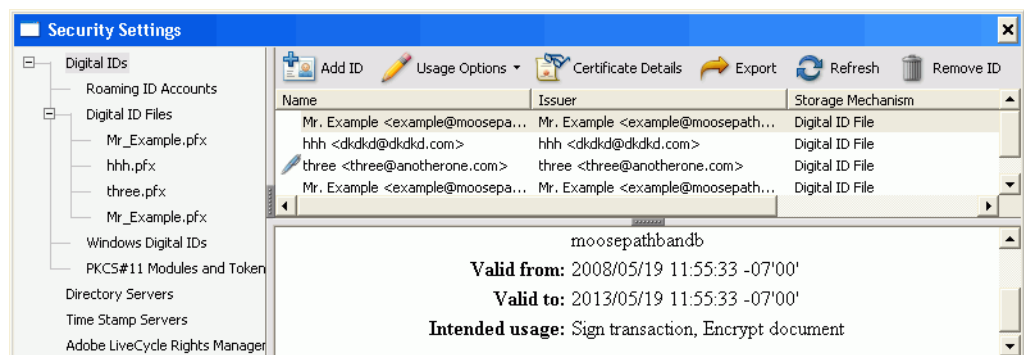
Older application versions use a deprecated digital ID format with an .apf extension. .apf is not supported in 9.0. You must use Acrobat 8.x or earlier to use this type of ID.

2.1.4 Digital ID Management and the Security Settings Console

The Security Settings Console enables users to manage their own digital IDs. Choosing **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings** opens a dialog for adding, removing, and setting the usage preferences for digital IDs stored on .pfx files, PKCS#11 modules and tokens, roaming ID servers, and the Windows Certificate Store.

Tip: You should always back up your private key if you have access to it. Without the key, encrypted documents cannot be decrypted and opened. To protect and back up private keys in an enterprise setting, administrators sometimes escrow private keys. If your digital ID is stored in a file on your local machine, consider copying it to a secure location.

Figure 4 Security settings menu and manager



2.1.5 Setting Identity Information

You can enter default identity (user) information that the application can automatically use as the defaults for workflows such as creating self-signed certificates and emailing certificate and server settings.

To create default user information:

1. Choose one of the following.
 - Acrobat (Windows): **Edit > Preferences > Identity**
 - Acrobat (Macintosh): **Acrobat > Preferences > Identity**
 - Adobe Reader (Windows): **Edit > Preferences > Identity**
 - Adobe Reader (Macintosh): **Adobe Reader > Preferences > Identity**
2. Configure the identity details. These details will appear in your signature appearance when you sign with a self-signed digital ID.
3. Choose **OK**.

Figure 5 Identity preferences

2.2 Generic ID Operations

Once you have one or more digital IDs, you can edit, remove, and otherwise manage them from the Security Settings Console. To simplify workflows that use digital IDs, consider doing the following before using your ID:

- [Specifying Digital ID Usage](#): Set an ID to automatically use each time one is required for signing or certificate encryption.
- [Sharing \(Exporting\) a Digital ID Certificate](#): Since a digital ID's certificate contains the public key required for validating your digital signature and encrypting documents for you, send it to those who participate in these kinds of workflows with you ahead of time.

Other operations also apply to all digital IDs irrespective of their format. For details, see:

- ["Viewing All of Your Digital IDs" on page 16](#)
- ["Customizing a Digital ID Name" on page 17](#)
- ["Viewing Digital ID Certificates in the Certificate Viewer" on page 17](#)

2.2.1 Specifying Digital ID Usage

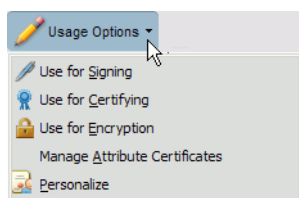
If a digital ID is not specified for a particular task that requires one, a prompt will ask for a digital ID file. To avoid repeated prompts, specify a digital ID for signing and encryption. Different IDs may be used for signing and encryption.

When you specify ID usage, that ID is the first one in the list you'll see when you're asked to select an ID in a signing or encryption workflow. If you select a different ID, your usage option will change to the newly selected ID; that is, the last used ID becomes the new "default."

To select a default digital ID file:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Select **Digital IDs** in the left-hand tree (Figure 2.2.1).
3. Highlight an ID in the list on the right.
4. Choose **Usage Options**. A drop-down list appears.

Figure 6 Usage options for a digital ID



5. Choose one or more options: signing, certifying, and encrypting. A lock or pen icon (or both) will appear to the left of the digital ID based on this selection.

Caution: Invalid and expired IDs with a yellow caution triangle cannot be used.

2.2.2 Sharing (Exporting) a Digital ID Certificate

Digital ID certificates must be distributed among participants in signing and certificate encryption workflows. Other users must have access to your certificate before:

- They can validate your signature if they are not already trusting a certificate above yours in the certificate chain. Note that a signature always includes the signer's certificate, so validation can occur with the certificate embedded in the signature if it is not already on the validator's machine.
- They can encrypt a document for you using certificate security.

Certificates can be emailed or saved to a file. You can also use FDF files to export your certificate so that others can import it into their trusted identities list. For details, see ["Exporting Your Certificate" on page 158](#).

Note: To export a certificate displayed in the Certificate Viewer, choose **Export** on the Summary tab.

2.2.3 Viewing All of Your Digital IDs

You can view all of your digital IDs in one list regardless of their type or location.

To view all of your IDs:

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

2. Select **Digital IDs** in the left-hand tree (Figure 4).

All the IDs you have added appear in the right hand panel. The list includes all of the IDs that you can view separately under:

- Digital ID Files
- Roaming ID Accounts
- Windows Digital IDs
- PKCS#11 Modules and Tokens

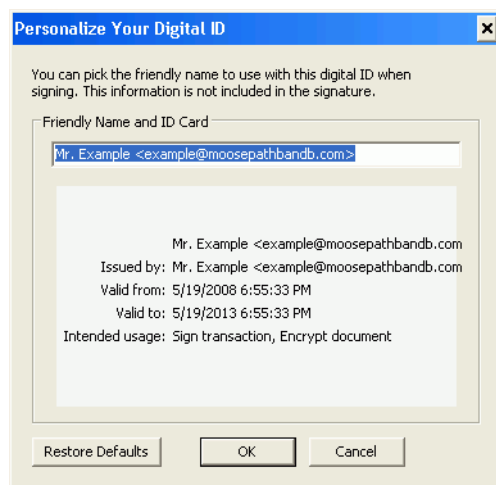
2.2.4 Customizing a Digital ID Name

You can personalize a digital ID by providing a user-friendly name. This name appears in the ID drop-down list in workflows where you are asked to select an ID.

To provide a friendly name:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Select **Digital IDs** in the left-hand tree (Figure 4).
3. Highlight an ID in the list on the right.
4. Choose **Personalize**.
5. Enter a name for the ID.

Figure 7 Personalizing an ID name



2.2.5 Viewing Digital ID Certificates in the Certificate Viewer

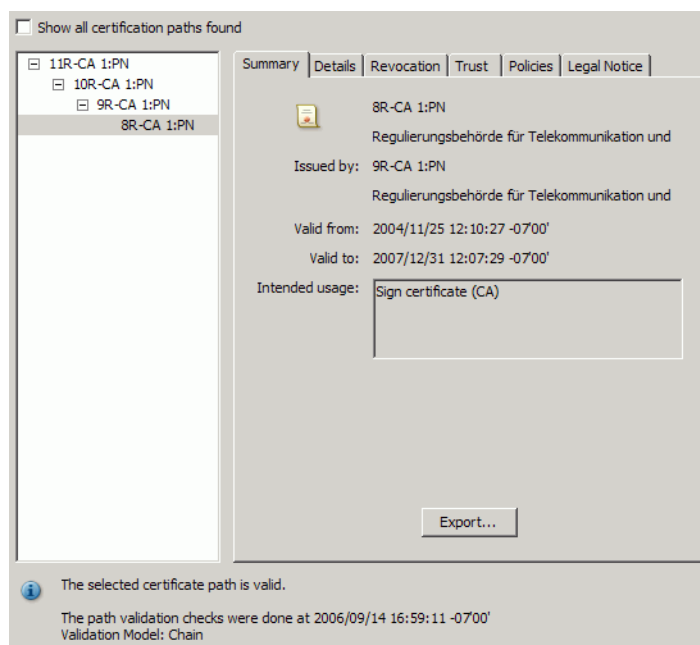
Your digital IDs appear in the Security Settings Console. From there, the Certificate Viewer can be used to display the time for which its certificate is valid and other details such as usage, a unique serial number, public key method, and so on (Figure 8).

To check certificate details:

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

2. Select **Digital IDs** in the left-hand tree ([Figure 2.2.1](#)).
3. Highlight an ID in the list on the right.
4. Choose **Certificate Details**. The Certificate Viewer displays the certificate. ([Figure 8](#)). The following details are available:
 - **Left hand panel:** The certificate chain.
 - **Bottom area:** A description of the certificate, path validity statement, path validation time, and sometimes the type of validation.
 - **Summary tab:** Displays the owner, issuer, validity period, and other details. The Intended Usage field tells you whether the certificate can be used for signing, encryption, or both. An **Export** button allow you to export the certificate to a file.
 - **Details tab:** Lists all the certificate fields (extensions) and their values.
 - **Revocation tab:** Indicates whether a revocation check occurred and the result. Allows users to initiate a manual check and analyze problems.
 - **Trust tab:** Displays the certificate's trust level. If it does not already exist in the trusted identities list, the **Add to Trusted Identities** is active. If the certificate is already on the Trusted Identities list and you want to change the trust level, see "[Certificate Trust Settings](#)" on page 35.
 - **Policies tab:** Displays policy restriction information that must be met for a signature to be valid, if any.
 - **Legal Notice tab:** Displays other certificate policies as well as a button which links to that policy, if any.

Figure 8 Digital ID: Certificate viewer

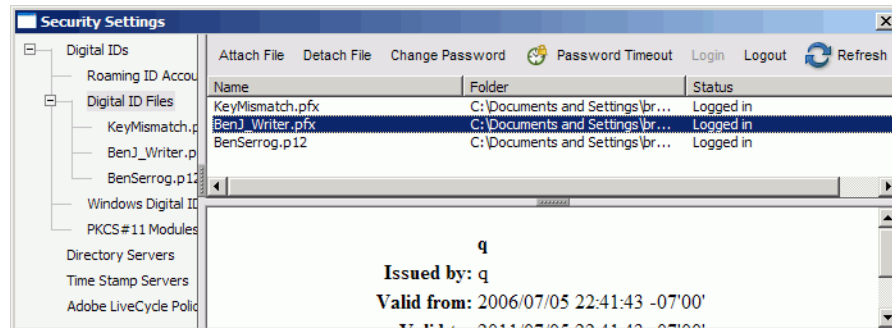


2.3 Managing PKCS#12 Digital ID Files

PKCS#12 digital ID files have several convenient features:

- Multiple IDs can be stored in a single, password-protected file.
- A file can contain both the public and private key.
- Passwords and password time-outs are user customizable.

Figure 9 Digital ID Files menu



2.3.1 Logging in to a Digital ID File

You will not usually need to log in to a digital ID file. Logging in means that Acrobat wants you to prove that you know the password to open the password-protected file containing the digital IDs. Since you likely supplied the password when you created your ID or obtained a new one, then you should be logged in.

However, you may need to log in for the following cases:

- You logged out of the file for some reason.
- You are importing an acrobatsecuritysettings file containing digital IDs.

To log in to a digital ID file:

2.3.2 Adding a Digital ID from a PKCS#12 File

If you need a digital ID does not appear in the digital ID list and you know it's location, browse to it and add it. You can browse to PKCS#12 files (with .pfx or .p12 extensions) and Windows Certificate Store compatible files (with .cer and .der extensions).

Note: In enterprise settings, you may be instructed by your administrator to get a digital ID from a specific location or to customize Acrobat or Adobe Reader to work with software supplied by your organization.

To find a digital ID file:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Select **Digital IDs** in the left-hand tree (Figure 4).

3. Choose **Add ID**.
4. Select the **My existing digital ID from** and **A file** radio buttons (Figure 3).
5. Choose **Next**.
6. Choose **Browse** and browse to the digital ID file. PKCS#12 files may reside on a network or in some local location. For example,
 - On a Window machine it might be C:\Documents and Settings\<username>\Application Data\Adobe\<application name>\<version>\Security\.
 - On a Windows machine with Vista in low rights mode and installed from a browser, it is <Boot Drive>:\Users\<user name>\AppData\Roaming\Adobe\<application name>\<version>\Security
7. Select the ID and choose **Open**.
8. Enter a password if one is required.
9. Review the digital ID list and choose **Finish**.

2.3.3 Adding and Removing Digital ID Files from the File List

Adobe Acrobat and Adobe Reader only allow deletion of user-created self-signed digital IDs created with those applications. A file can have one or more IDs.

To delete or add an ID file:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Select **Digital IDs > Digital ID Files** in the left-hand tree (Figure 9).
3. Highlight a digital ID file in the right-hand panel.
4. Do one of the following:
 - Choose **Detach File**. The file is removed from the list but still remains on your file system.
 - Choose **Attach File**. Browse to the file, enter the file password, and choose **OK**.

Note: Detaching a file does not remove it from your system, and it may be reattached later.

2.3.4 Changing an ID File's Password

Passwords and password time-outs are unique to PKCS#12 IDs. Since a file can contain multiple IDs, passwords and time-outs are configured at the file level rather than for individual IDs.

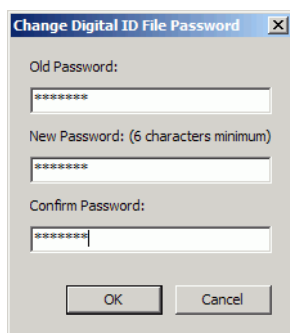
Note: If the file is read only, then the **Change Password** and **Password Timeout** options are disabled.

To change the password:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Highlight **Digital ID Files** in the left-hand tree (Figure 9).
3. Select a file in the right-hand panel (Figure 9).

4. Choose **Change Password**.
5. Enter the old password.
6. Enter a new password and confirm it.
7. Choose **OK**.

Figure 10 Digital ID files: Password configuration



2.3.5 Changing a PKCS#12 File's Password Timeout

Passwords and password time-outs can only be set for PKCS#12 IDs. Since a file can contain multiple IDs, passwords and time-outs are configured at the file level rather than for individual IDs.

Note: If the is read only, then the **Change Password** and **Password Timeout** options are disabled.

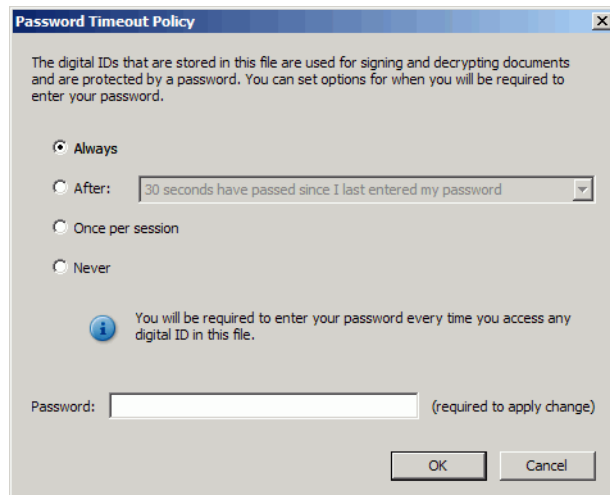
To change the password timeout:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Highlight **Digital ID Files** in the left-hand tree (Figure 9).
3. Select a file in the right-hand panel (Figure 9).
4. Choose **Password Timeout**.

Tip: The password timeout feature interacts with the Login/Logout feature as described in [“Logging in to PKCS#12 Files” on page 22](#).

5. Configure the Password Timeout Policy dialog by specifying when a password prompt should appear:
 - **Always:** A password is always required each time the digital ID is used regardless of whether or not you are logged in to a file.
 - **After:** Choose a value from the drop-down list to set a time frame.
 - **Once per session:** A password is asked for only once while the application is open.
 - **Never:** The password is not usually required when using this ID and you are logged into the file.
6. Enter the password.
7. Choose **OK**.

Figure 11 Digital ID files: Timeout settings



2.3.6 Logging in to PKCS#12 Files

The digital ID Login feature provides access to the IDs in a particular file. Login behavior is dependant on the user-specified password timeout feature. If the user has specified a password timeout of **Never**, then the application never asks for a password when an ID is used for some process. For example:

- **Signing:** During signing workflows, you can sign with a digital ID without entering a password if you are logged into a file and the time-out is set to **Never**.
- **Batch processing:** In normal operation, batch sequences that require access to a digital ID invoke the user-interface's authentication dialog. Because the dialog prompts for a password, the batch sequence is effectively stopped until a user intervenes. Logging in to a file provides the ID to the process without stopping it or requiring user input.

To enable sequences to run automatically and bypass normal user interface actions, do the following:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Select **Digital ID Files** in the left-hand tree (Figure 9).

Tip: Verify the password timeout is set according to your own preferences. For details, see [“Changing a PKCS#12 File’s Password Timeout” on page 21](#).

3. Select a file in the right-hand panel (Figure 9).
4. Do one of the following:
 - **Logout:** Highlight an ID in the list on the right and choose **Logout**.
 - **Login:** Highlight an ID in the list on the right and choose **Login**. Enter a password when prompted and choose **OK**.

2.3.7 Creating a Self-Signed Digital ID

Note: The option to create self-signed digital IDs is unavailable if your administrator has configured your application to prevent this operation.

Users can create a self-signed digital ID if they don't wish to purchase an ID from a 3rd party certificate authority (CA) or are not given a company-provided ID. Self-signed IDs are usually considered less secure because the user has not been verified by a 3rd party CA. For self-signed IDs, you act as your own CA.

To create a self-signed digital ID:

1. Navigate to the Add Digital ID dialog as described in [“Adding a Digital ID from a PKCS#12 File” on page 19](#).
2. Choose **A new digital ID I want to create now** (Figure 3).
3. Choose **Next**.

Figure 12 Digital ID format selection



4. Select a digital ID format and storage location:
 - **New PKCS#12 Digital ID File:** Stores the IDs in a password protected file with a .pfx (Win) or .p12 (Mac) extension. The file is in a PKCS#12 standard format. The files can be copied, moved, and emailed. They are cross-platform, portable, and always password protected. This common format is supported by most security software applications, including web browsers. These files should always be backed up. On Windows XP, the default location is `C:\Documents and Settings\<username>\Application Data\Adobe\<application name>\<version>\Security\`.
 - **Windows Certificate Store:** (Windows only) Stores the ID in the Windows Certificate Store where it is also available to other Windows applications. The ID is protected by your Windows login. These IDs are easy to use and do not have to have file-level password protection. However, they are not portable and could be less secure if a file-level password is not specified.
5. Choose **Next**.

Figure 13 Digital ID: Configuration

Add Digital ID

Enter your Identity information to be used when generating the Self-Signed Certificate.

	ASCII	Unicode
Name (e.g. John Smith):	Neb Studly	
Organizational Unit:	Aardvark	
Organization Name:	AntEater Inc.	
Email Address:	nebrogeros@anteater.com	
Country/Region:	VA - HOLY SEE (VATICAN CITY STATE)	
<input checked="" type="checkbox"/> Enable Unicode Support		
Key Algorithm:	1024-bit RSA	
Use Digital ID for:	Digital Signatures and Data Encryption	

Cancel < Back Next >

6. Configure the digital ID. The dialog is prepopulated if the Identity preferences have been previously configured:

Tip: If you use non-Roman characters, choose **Enable Unicode Support** before continuing.

- **Name:** The name that appears in the Signatures tab and in the signature field.
- **Organizational Unit:** Optional. Appears in the signature and certificate.
- **Organizational Name:** Optional. Appears in the signature and certificate.
- **Email Address:** Optional. Appears in the signature and certificate.
- **Country/Region:** Optional. Appears in the signature and certificate.
- **Enable Unicode Support:** Optional. Use Unicode when your information cannot be adequately displayed with Roman characters.

Note: Many applications do not support non-ASCII characters in certificates. Be sure to specify both an ASCII representation of the information as well as the Unicode representation of information you are supplying.

- **Key Algorithm:** 2048-bit RSA offers more security than 1024-bit RSA, but 1024-bit RSA is more universally compatible. Use the 1024 bit key length if you are unsure.
 - **Use Digital ID for:** Select whether to use the digital ID for digital signatures, data encryption (certificate security), or both.
7. If a Windows digital ID was selected, choose **Finish**; otherwise, for a PKCS#12 ID do the following:
 1. Choose **Next**.
 2. Specify a file name and location for the digital ID file.
 3. Enter a password and confirm it.

Note: Passwords are case-sensitive and must contain at least six characters.

4. Choose **Finish**.

Figure 14 Digital ID: PKCS#12 location and password

Enter a file location and password for your new digital ID file. You will need the password when you use the digital ID to sign or decrypt documents. You should make a note of the file location so that you can copy this file for backup or other purposes. You can later change options for this file using the Security Settings dialog.

File Name:

Password:

Confirm Password:

2.3.8 Deleting a PKCS#12 Digital ID

Adobe Acrobat and Adobe Reader only allow deletion of user-created, self-signed digital IDs created by them. The methodology for deleting other types of IDs varies with the type of ID.

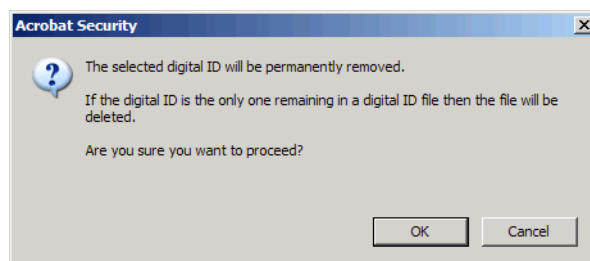
While the ID will be removed from the ID list, other ID's in the container .pfx or p12 file will not be affected. Deleting the last, self-signed PKCS#12 ID in a .pfx or p12 file also deletes the digital ID file.

Caution: Because deleting an ID deletes its private key, operations that require that key will no longer be possible. If the file is used by other programs or you need it to open encrypted documents, do not delete it.

To delete a self-signed ID:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Select **Digital IDs** in the left-hand tree ([Figure 2.2.1](#)).
3. Highlight a self-signed ID in the list on the right that uses a digital ID file or Windows Certificate Store storage mechanism.
4. Choose **Remove ID**.
5. Choose **OK** when asked to proceed.

Figure 15 Digital ID: Deleting

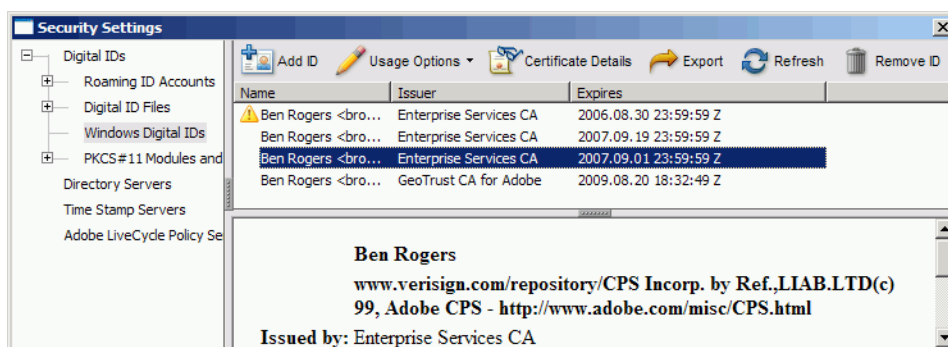


2.4 Managing Windows Digital IDs

For the Acrobat family of products, a “Windows digital ID” is an ID that resides in the Windows certificate store rather than the Acrobat store. Windows supports several formats listed in [Table 2](#). These IDs are protected by your Windows login, are easy to use, and do require file-level password protection. However, they are not portable and are less secure when a file-level password is not specified.

The Windows store makes these IDs available to other Windows applications such as Acrobat and Adobe Reader. When an ID in the Windows store is registered with the application, it appears in the Security Settings Console. IDs in the Windows store are subject to the same operations as described in [“Generic ID Operations” on page 15](#)

Figure 16 Windows digital ID menu



2.4.1 Finding a Digital ID in a Windows Certificate Store File

If you have a personal digital ID in the Windows store, it should appear in the Security Settings Console automatically without any special configuration. Acrobat products automatically find that ID. However, if there is a problem, you can browse to and add Windows Certificate Store compatible files (.cer and .p7b).

2.4.2 Deleting a Digital ID from the Windows Certificate Store

IDs that have been added to the Windows certificate store can only be deleted from the Security Settings Console if they are self-signed IDs created in Acrobat or Reader version 8.0 or later. Other IDs must be removed from the Windows store by using an application such as Internet Explorer. The store's location in Internet Explorer may vary by version, but is typically found under **Tools > Internet Options > Content tab > Certificates button**.

Roaming IDs allow you to access and use your digital ID for signing or encryption from any machine that can access the server. You don't have to have your ID file with you or install it prior to use.

Note: Roaming IDs can be centrally administered. When IDs expire, new ones can be issued and placed on the server rather than being distributed to each individual. Deployment and management therefore occurs in one location rather than on numerous client machines.

- [“Security Setting Import and Export” on page 149](#)

6.

7.

2.5 Your server may require additional or different authentication steps. Follow directions that appear in the dialogs.

Managing IDs Stored on Hardware Devices

Smart cards, hardware tokens, and other devices are increasingly being used by businesses and individuals to carry digital IDs. These devices provide enhanced mobility, remote access to intranets and extranets, as well as strong security with public/private key cryptography and PIN access to the digital ID.

Note: Most devices comply with the Public Key Cryptography System 11 (PKCS#11) format devised by RSA.

The method for registering a digital ID on such a device with the application may vary. The manufacturer or your system administrator should provide detailed instructions. However, the steps below may be used as a general guide. IDs stored on a PKCS#11 device are subject to the same operations as described in [“Generic ID Operations” on page 15](#).

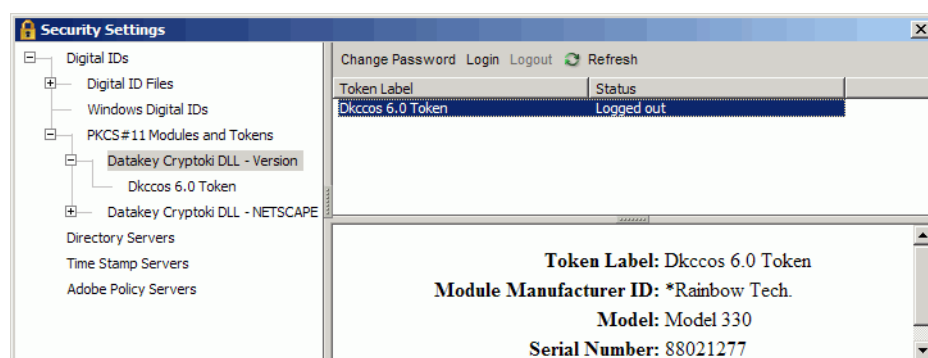
2.5.1 Adding an ID that Resides on External Hardware

Digital IDs can reside on hardware such as a smart card or token with a USB interface. In these cases, the card is inserted into a smart card reader or the token is inserted directly into a USB port. Adobe products can be configured to look for and use IDs on these devices by adding the device's module (software driver) to the module list. The module's IDs are automatically registered with the application.

To register an ID that resides on external hardware:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Expand **Digital IDs** in the left-hand list ([Figure 16](#)).
3. Highlight **PKCS#11 Modules and Tokens**.

Figure 17 PKCS#11 Security Settings menu items



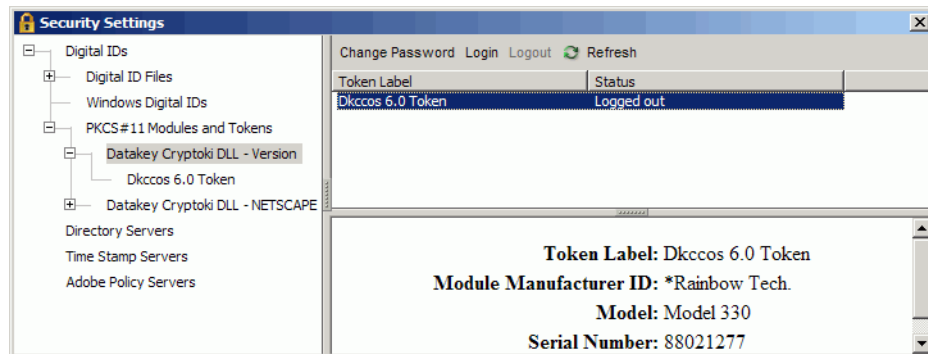
4. Choose **Add Module**.
 5. Browse to the device driver. On Windows, this could likely be C:\Windows\system32\<some dll>.dll. The exact path will be supplied by your system administrator or the maker of your device.
 6. Choose **Open**.
- The module and its IDs are automatically added to the list in the right-hand panel.

2.5.2 Changing Passwords

A card or token may contain multiple IDs. All of the IDs are password protected by a single password. This password is used to log in to a device and to sign.

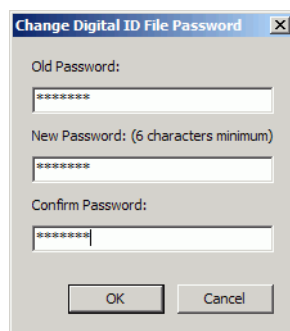
1. Expand the tree under **PKCS#11 Modules and Tokens**.
2. Highlight any module.

Figure 18 PKCS#11 Security Settings menu items



3. A card or token label should appear in the right-hand panel. If there is more than one, select one.
4. Choose **Change Password**.
5. Enter the old password.
6. Enter a new password and confirm it.
7. Choose **OK**.

Figure 19 Digital ID files: Password configuration



2.5.3 Logging in to a Device

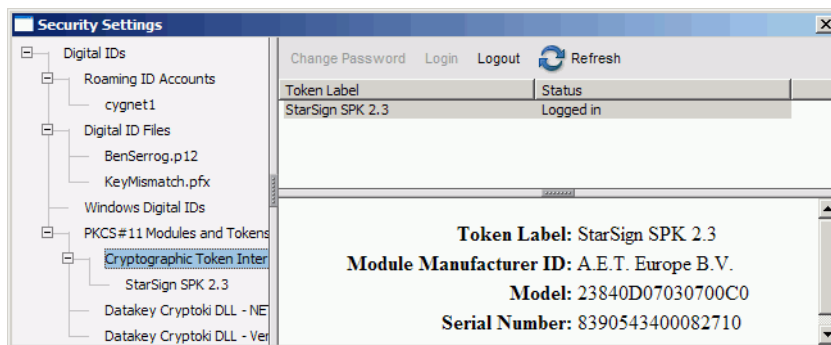
Logging in provides access to the IDs on a particular device or smart card. In most cases login in is not required as it occurs on demand during signing or encryption/decryption.

PKCS#11 workflows vary by the device supplier. For example, additional passwords or PINs may or may not be required. The login interface may be provided by the Adobe application or by the device supplier.

To log in to a device:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Expand the tree under **PKCS#11 Modules and Tokens**.
3. Highlight any module.
4. A card or token label should appear in the right-hand panel. If there is more than one, select one.
5. Choose **Login**.
6. Enter a password.
7. Choose **OK**.

Figure 20 PKCS#11 Security Settings menu items



As described in [“What is a Digital ID?” on page 11](#), a digital ID consists of two main parts: a certificate with a public key and a private key. Participants in signing and certificate security workflows need to exchange the public part (the certificate) of their digital ID. Once you obtain someone’s certificate and add it to your trusted identities list, you can encrypt documents for them. If their certificate does not already chain up to a trust anchor that you have specified, you can set the certificate’s trust level so that you can validate the owner’s signature.

Understanding what a trusted identity is and how trust levels are set can help you set up streamlined workflows and troubleshoot problems. For example, you can add trusted identities ahead of time and individually set each certificate’s trust settings. In enterprise settings where certificates are stored on a directory server, you may also be able to search for certificates to expand your list of trusted identities.

For more information, refer to the following:

- [“What is a Trusted Identity?” on page 30](#)
- [“Using Directory Servers to Add Trusted Identities” on page 38](#)
- [“Adding Someone to Your Trusted Identity List” on page 32](#)
- [“Managing Contacts” on page 42](#)

3.1 What is Trust?

The concept of “trust” is complex, and it may mean different things in different contexts. In Acrobat security workflows, trust can mean the following:

- **Trusting participants in your workflows:** In both document security and signature workflows, you will need to trust those with whom you are sharing your documents. “Trusting an identity” means that you accept that someone’s certificate actually represents a particular person or organization. It is official recognition on your part of the ownership and origin of the digital ID; that is, that the digital ID represents a specific entity.
- **Setting certificate trust levels:** For those in your list of trusted identities, you will likely need to allow and disallow certain operations. You do this by associating (setting) trust levels with trusted identity’s certificate. These trust levels define privileges that allow documents signed or certified by that identity to execute privileged operations on YOUR machine--things that cannot otherwise be done by documents you otherwise just open and display--for example, playing multimedia or executing JavaScript. Providing trust to a certificate should only be done if it is necessary and you want documents created and signed by the trusted identity to have higher levels of access to your machine.

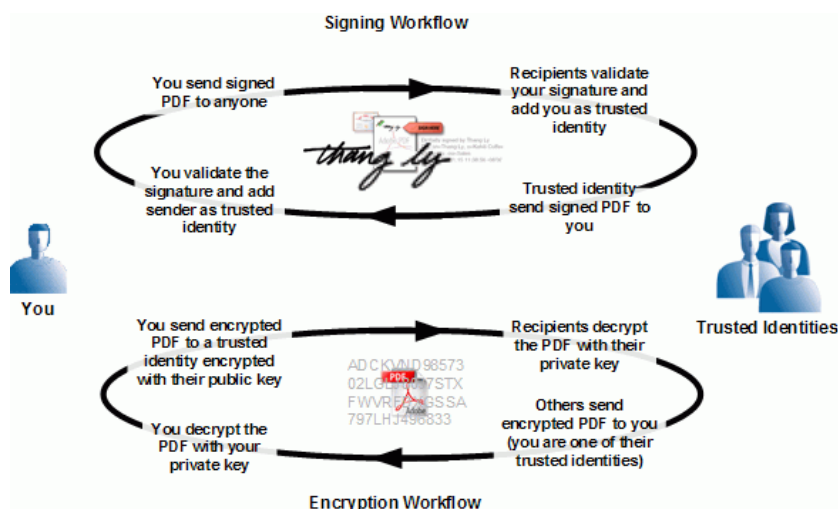
3.2 What is a Trusted Identity?

Digital signature and certificate security workflows both rely on certificates. Participants in signing workflows share their certificates ahead of time or embed them in a document. Participants in certificate

security workflows must share their certificates ahead of time. Both operations involve importing other people's certificates into your Trusted Identities list. When a person's certificate information appears in the Trusted Identity Manager, they become a *trusted identity*.

Groups of people that share documents with certificate security or digital signatures are in essence a community of trusted identities that share their certificates to make those features work. You will add people to your trusted identity list and others will add you to theirs:

- When you sign document, the document recipient can validate your signature by validating the certificate embedded in the document. Conversely, you need access to a document sender's certificate to validate their signature.
- You encrypt a document with the document recipient's public key so that they can decrypt it with their corresponding private key. Conversely, others need your certificate to encrypt documents for you.

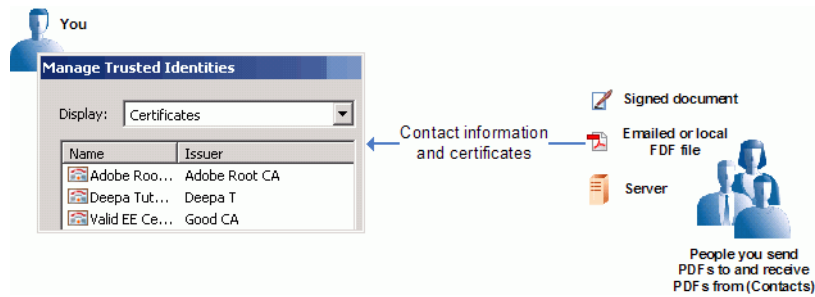


The Acrobat family of products provide tools for selecting and interacting with the certificates of document recipients you trust. For example, Acrobat's user interface prompts authors to select one or more recipients when applying certificate security. Because it is often the case that a document will be sent or received from numerous individuals, it is expedient to create a list of trusted identities ahead of time. In large organizations, an administrator may do this for you; otherwise, you will use Acrobat's Trusted Identity Manager to store your trusted identities' contact information and certificates.

Getting someone's contact information and certificate involves searching for (or having sent to you) the digital ID data in the requisite format. Some common ways of getting the data include the following:

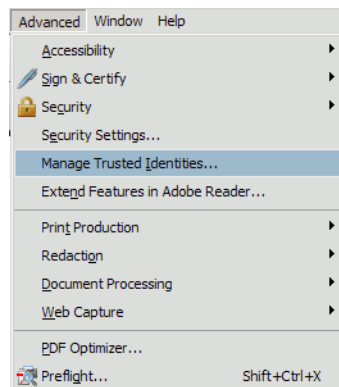
- **Import the data from an .acrobatsecurity file.** Configuration details can be imported from a security settings file as described in ["Migrating and Sharing Security Settings" on page 149](#).
- **Extract the data from an FDF file.** Double-clicking on an FDF file causes Acrobat to automatically import the information.
- **Search a server directory.** Users can add directory servers containing contact information and certificates. Sometimes administrators preconfigure these directories.
- **Use the data embedded in a signed document.** The Certificate Viewer's **Add to Trusted Identities** button adds a certificate to the trusted identities list and allows setting its trust level.

Figure 21 Digital ID: Managing trusted identities



From within the Manage Trusted Identities dialog, users import and manage the certificates and certificate owner data for document recipients they wish to trust. A contact will occasionally be associated with multiple certificates. Therefore, contacts and certificates are in some respects managed independently of each other. It is also possible to create a group from any number of contacts so that security can be applied to all group members with a single action. Users manage contacts, groups, and certificates by choosing **Advanced** (Acrobat) or **Document** (Reader) > **Manage Trusted Identities** and opening the Trusted Identities Manager.

Figure 22 Manage Trusted Identities menu item



3.3 Adding Someone to Your Trusted Identity List

As shown in [Figure 21](#), you build a list of trusted identities by getting digital ID certificates from those who will be participating in signing and certificate security workflows. You get this information from a server, a file, or from a signed document. For signing workflows, you can get this information during the signature validation process. For certificate security workflows involving encryption, you must request the information ahead of time so you can encrypt the document with the document recipient's public key.

Figure 23 Certificate Viewer: Trust tab

3.3.1 Requesting a Digital ID via Email

Email requests for digital ID information use .acrobatsecurity or FDF files. For details, see [“Migrating and Sharing Security Settings” on page 149](#).

For details, see [“Requesting a Certificate via Email” on page 161](#).

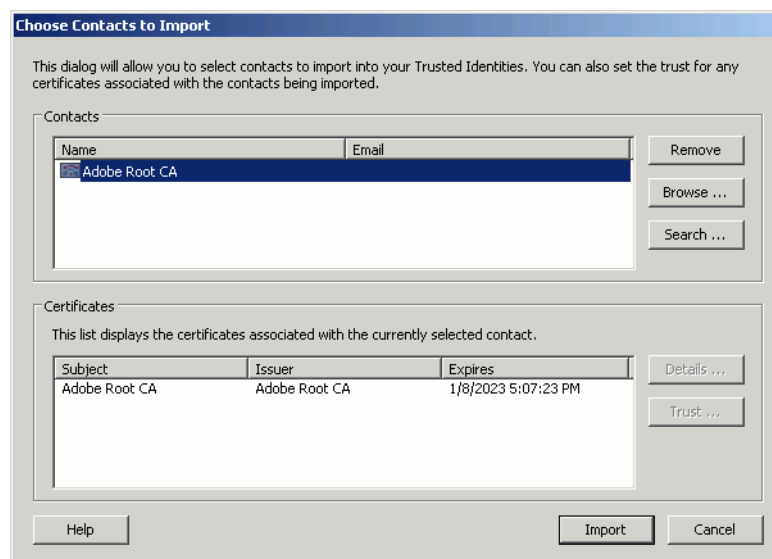
3.3.2 Importing a Certificate From a File

Acrobat and Adobe Reader can export certificates to a file so that they can be shared as needed. To import certificates, follow the instructions described in [“Migrating and Sharing Security Settings” on page 149](#).

However, certificates may also exist in other file types such as .cer, .p7b, and so on. To import certificates from these file types:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Manage Trusted Identities**.
2. Choose **Add Contacts**.
3. Choose **Browse**.
4. Browse to the contact file location.
5. Select the file.
6. Choose **Open**.

Figure 24 Importing digital ID data



7. Choose **Import**.
8. Choose **OK** when the confirmation dialog appears.

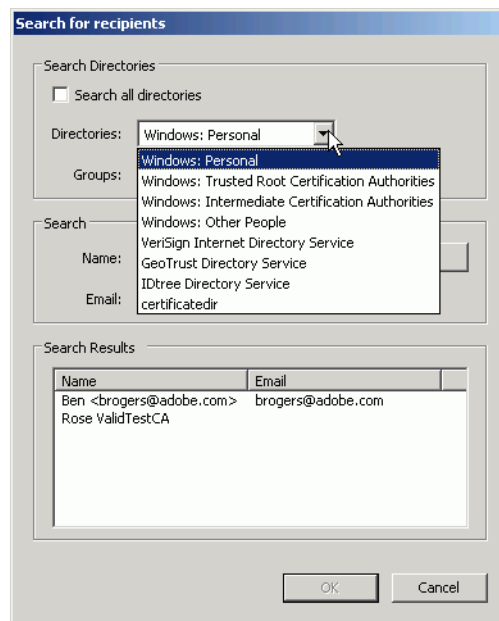
3.3.3 Searching for Digital ID Certificates

The search feature allows you to search a list of directories for certificates. If no directories have been previously specified, the **Search** button will NOT appear. The list of search servers in the Directories drop-down list is populated through three mechanisms:

- The default server settings that ship with Adobe Acrobat and Adobe Reader.
- The Windows Certificate Store if the user has turned on this option.
- User-specified directory servers the user has added in the Security Settings Console. For details, see ["Using Directory Servers to Add Trusted Identities" on page 38](#).

Tip: Home users do not usually need to change the directory server list. Users in enterprise environments typically have the list preconfigured by their system administrator.

Figure 25 Digital IDs: Searching for certificates

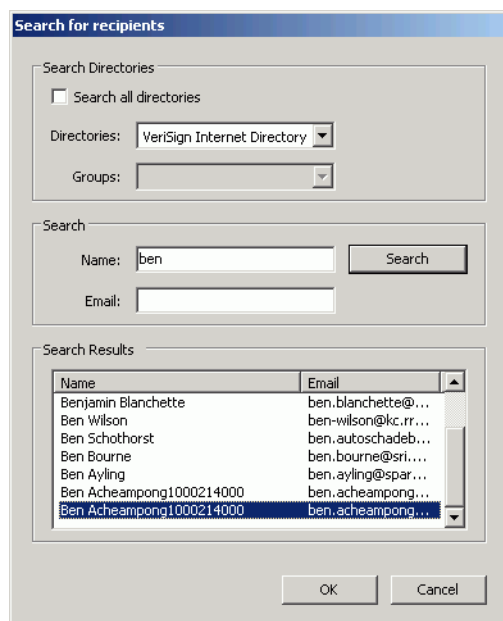


To search for a certificate so that you can add one or more people to your trusted identities list:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Manage Trusted Identities**.
2. Choose **Add Contacts**.
3. Choose **Search**.
4. Configure the search options:
 - Choose **Search all directories** or select a directory and optional group.
Searching all directories may take some time. In a business environment, it is often expedient to just select the company's LDAP directory.
 - Enter a name and/or email address to search. This is an AND search. Using both fields only returns results that match both criteria.
5. Choose **Search**.

6. Select a name from the search results.
7. Choose **OK**.
8. If the desired entries are found, choose **Import**.
9. Choose **OK** when the confirmation dialog appears.

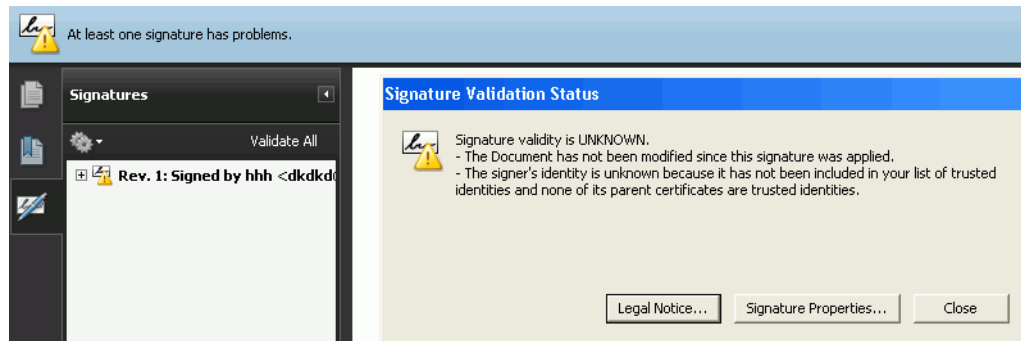
Figure 26 Searching for a document recipients



3.4 Certificate Trust Settings

Contacts in the trusted identities list should be associated with one or more certificates. Those certificate's trust settings may be individually configured. Choosing to not trust a certificate does not prevent a document from displaying, but it will result in signatures having an problematic status. The status is represented by a yellow triangle in the Document Message Bar, Signatures pane, and the Signature Validation Status dialog (Figure 27). For each contact for whom you will encrypt a document with certificate security, one certificate can also be selected as the default for encryption.

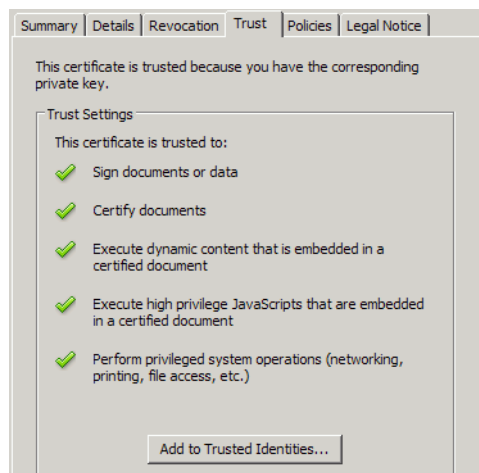
Figure 27 Untrusted signature



Certificate trust settings have the following features:

- Trust settings are configured in the Trusted Identity Manager ahead of time, at the time of import, or directly from a signature.
- Trust settings can be viewed in the Trusted Identity Manager by choosing **Edit Trust** or by choosing the Trust tab in the Certificate Viewer (Figure 28).
- Certificates can be separately trusted for approval signatures and certification signatures.
- Certificates can be individually configured to trust operations such as signing, certification, and allowing items such as dynamic content and JavaScript in certified documents. These settings interact with application environment settings.

Figure 28 Certificate trust settings



1. Do one of the following:
 - If you already have the certificate:
 1. Choose **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Manage Trusted Identities**.
 2. Choose **Certificates** in the **Display** drop down list.
 3. Select the certificate.
 4. Choose **Edit Trust**.
 - If the certificate is in a signature:

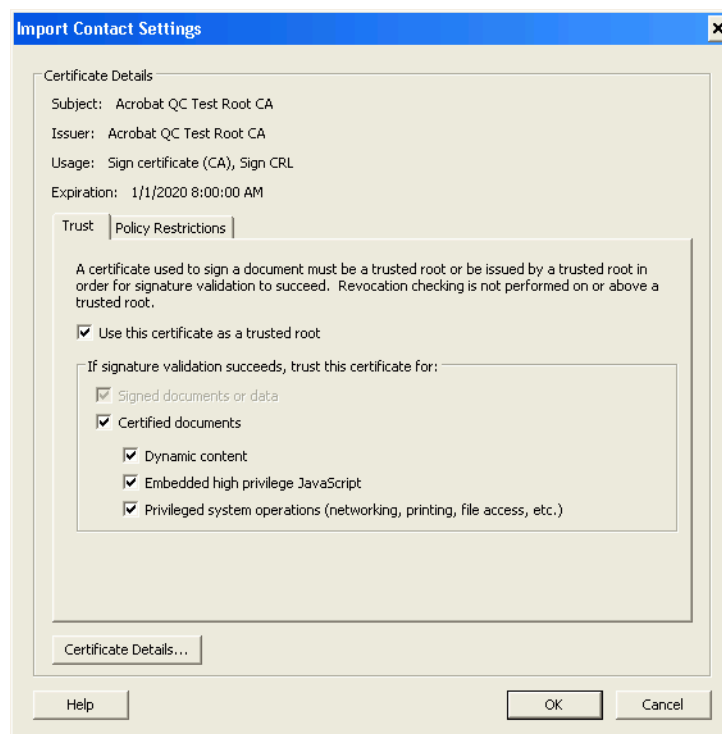
1. Right click and choose **Signature Properties**.
2. Choose **Show Certificate**.
3. Select the Trust tab.
4. Choose **Add to Trusted Identities**.

Tip: If **Add to Trusted Identities** is disabled, the identity is already on your Trusted Identities list. To change the trust settings, you must use the first method above.

2. On the Trust tab, select the trust options. In enterprise settings, an administrator should tell you which trust settings to use.

Note: During an import action, recipients of the distributed trust anchor may be able to inherit its trust settings. Once you've verified the sender, you usually want to accept these settings so you can use the certificate they way the sender intended.

Figure 29 Certificate trust settings



- **Use this certificate as a trusted root:** Makes the certificate a trust anchor. The net result is that any certificates which chain up to this one will also be trusted for signing. At least one certificate in the chain (and preferably only one) must be a trusted root (trust anchor) to validate signatures and timestamps.

Tip: There is no need to make end entity certificates trust anchors if they issued by a certificate holder whose certificate you have configured as a trust anchor. It is best practice to trust the topmost certificate that is reasonable to trust because revocation checking occurs on every certificate in a chain until that anchor is reached. For example, in a large organization, it is likely you would want to trust your company's certificate. If that

certificate was issued by VeriSign, you would not want to make VeriSign a trusted root unless you wanted to trust every certificate that chains up to VeriSign.

- **Signed documents or data:** Trusts the certificate for approval signatures.

Tip: This setting is disabled because if the certificate is set as a trust anchor. Trust anchors are automatically trusted for approval signatures.

- **Certified documents:** Trusts the certificate for certification signatures.
 - **Dynamic content:** Trusts multimedia and other dynamic content in certified documents. Selecting this option automatically adds documents that are certified with this certificate to the Trusted Documents list which is maintained by the Multimedia Trust Manager. For this reason, verify your application environment is configured correctly. For details, [“Controlling Multimedia” on page 136](#).
 - **Embedded high privilege JavaScript:** Trusts embedded scripts. Certificate settings do not override application-level settings, so even if JavaScript is enabled for a particular certificate, it may not execute unless the application’s preferences allow it. This option requires that the application environment be configured correctly. For details, see [“Setting JavaScript Options” on page 139](#).
 - **Privileged system operations (networking, printing, file access, etc.):** Some operations represent a security risk more serious than others. Acrobat considers the following operations potential threats to a secure application operating environment: Internet connections, cross domain scripting, silent printing, external-object references, and FDF data injection. If this checkbox is checked, documents that are certified with this certificate will allow these actions.

Tip: This feature interacts with the Enhanced Security preferences which may be set by choosing **Edit > Preferences > Security (Enhanced)**. The application always takes the least restrictive setting when determining what is allowed. For example, if the trust level for this certificate does not allow privileged operations but the certified file resided in a privileged location, then these operations will be permitted.

3. If you need to specify a policy restriction, do so. Most users only need to set policy restrictions at the request of their administrator. [“Setting Certificate Policy Restrictions” on page 39](#).
4. Choose **OK** twice.
5. Choose **Close**.

3.4.1 Using Certificates for Certificate Security (Encryption)

You only need to specify a certificate’s encryption usage if you are using certificate security. When more than one certificate is associated with the contact, you can select which one to use as the default encryption certificate. For details, see “Certificate Security” in the *Document Security User Guide*.

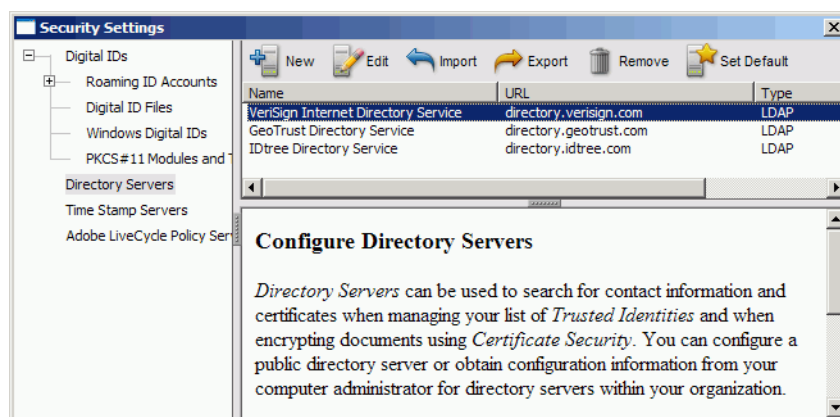
3.5 Using Directory Servers to Add Trusted Identities

Businesses often use a centrally managed certificate repository such as an LDAP directory server. Directory servers are capable of returning X.509 public key certificates. These servers are searchable so that you can easily expand your list of trusted identities. Both Adobe Acrobat and Adobe Reader for Windows ship with default servers:

- Versions 7.x:
 - VeriSign Internet Directory Service
 - GeoTrust Directory Service
 - IDtree Directory Service
- Version 8.x and 9x:
 - VeriSign Internet Directory Service

Home users may never need to use directory servers. In most cases, needed certificates will be sent directly to you or will be embedded in a signature. However, enterprise users will likely use directory servers when their administrator has set up an LDAP server as part of a public key infrastructure. This allows the administrator to make the certificates available to teams and workgroups while managing them from a central location. The administrator usually preconfigures user machines, tells the user how to configure the server manually, or sends the server configuration details in a file as described in [“Migrating and Sharing Security Settings” on page 149](#).

Figure 30 Digital ID Directory servers: Server list



3.5.1 Manually Configuring a Directory Server

Some companies store employee digital ID certificates on a networked LDAP server. To access those certificates, add the server to the list of directories used to locate those IDs.

Tip: In an ideal scenario, the server administrator supplies configuration details in a file as described in [“Migrating and Sharing Security Settings” on page 149](#).

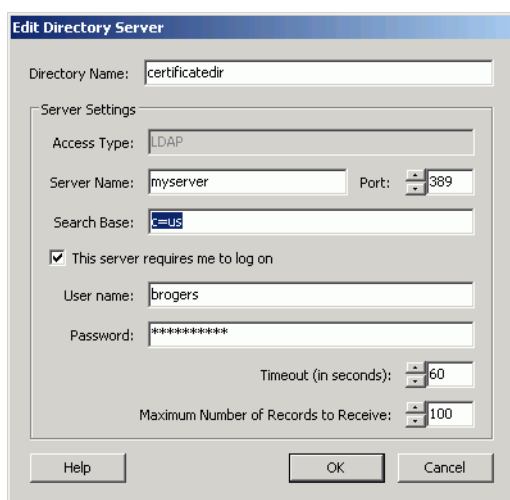
To manually configure an identity directory:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Select **Directory Servers** in the left-hand list ([Figure 30](#)).
3. Choose **New**.
4. Configure the LDAP server settings in the Edit Directory Server dialog:
 - **Directory Name:** An arbitrary directory name.
 - **Access Type:** LDAP is the only type supported.
 - **Server Name:** The server name.

- **Port:** The server port. 389 is the default port.
- **Search Base:** A comma-separated list of name-value pairs used in the search. For example, `c=us,cn=Brown Trout,ou=example,dn=Acme Manufacturing` for country, common name, organizational unit, and distinguished name.
- **This server requires me to log on:** Check this box if the server requires username and password authentication to look up LDAP entries.
- **User name:** The login username.
- **Password:** The login password.
- **Timeout:** The number of seconds to keep trying to connect.
- **Maximum Number of Records to Receive:** The number of records to return.

5. Choose **OK**.

Figure 31 Digital ID Directory servers: Setting server details



3.5.2 Editing Directory Servers Details

Directory server details can be changed at any time.

To edit directory server information:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Select **Directory Servers** in the left-hand list (Figure 30).
3. Select a directory server from the right-hand panel.
4. Choose **Edit**.
5. Edit the information as described in [“Manually Configuring a Directory Server” on page 39](#).
6. Choose **OK**.

3.5.3 Deleting a Directory Server

Previously configured directory servers can be removed from the server list at any time.

To delete a directory server:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Select **Directory Servers** in the left-hand list (Figure 30).
3. Select a directory server from the right-hand panel.
4. Choose **Remove**.
5. When a confirmation dialog appears, choose **OK**.

3.5.4 Specifying a Default Directory Server

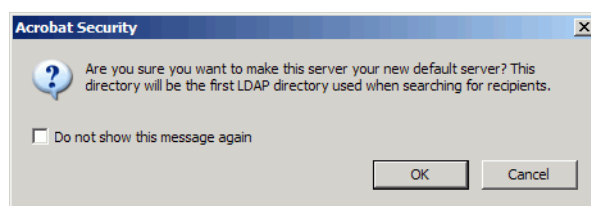
A default server may be specified so that it is always used when searching for digital IDs.

To set default directory server:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Select **Directory Servers** in the left-hand list (Figure 30).
3. Select a directory server from the right-hand panel.
4. Choose **Set Default**.
5. Choose **OK** if a confirmation dialog appears.

A star appears next to the name of the selected server.

Figure 32 Digital ID Directory servers: Setting defaults



3.5.5 Importing and Exporting Directory Server Settings

For details, refer to the following:

- ["Importing Directory Server Settings" on page 171](#)
- ["Emailing Server Details" on page 162](#)
- ["Exporting Server Details" on page 163](#)

3.6 Managing Contacts

Contacts are those people that will send you documents or receive documents from you. Each contact may be associated with one or more certificates. Like certificates, contacts can be added, removed, edited, and so on from the trusted identity list.

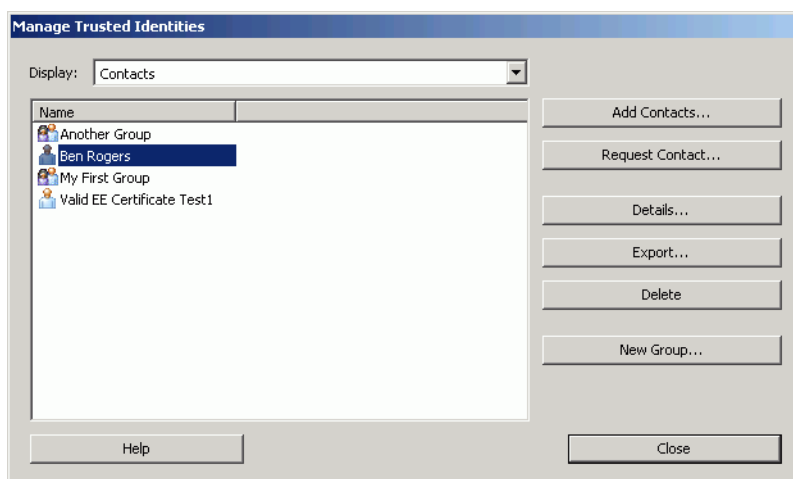
3.6.1 Viewing and Editing Contact Details

When a contact's details change, it is possible to update them in the Trusted Identity Manager.

To change a contact's details:

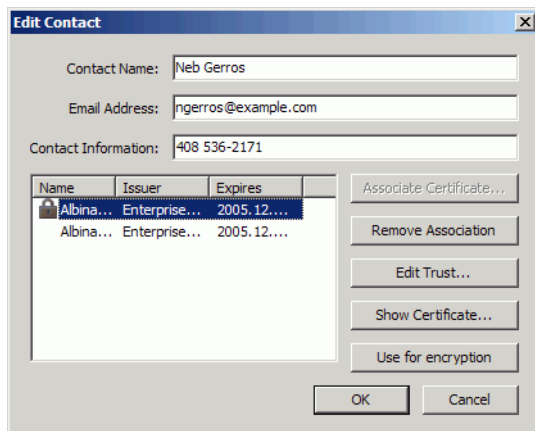
1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Manage Trusted Identities**.
2. Choose a contact in the left-hand list.

Figure 33 Contacts: Viewing details



3. Choose **Details**.

Figure 34 Edit Contact dialog



4. Edit the details.

5. Choose **OK**.

3.6.2 Emailing Certificate or Contact Data

You can export certificate and contact data via email directly from the Trusted Identity Manager. Doing so allows other users to add that data their trusted identity list, thereby expanding the number of users that can participate in secure document workflows. For details, see [“Emailing Your Certificate” on page 159](#).

3.6.3 Saving Certificate or Contact Details to a File

You can export certificate and contact data and save it to a file from the Trusted Identity Manager. Doing so allows you to email it later or locate it on a shared network directory. Other users can then add that data to their trusted identity list. For details, see [“Saving Your Digital ID Certificate to a File” on page 160](#).

3.6.4 Associating a Certificate with a Contact

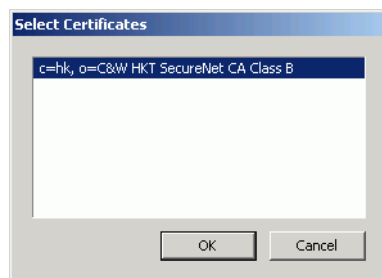
A certificate is usually already associated with a contact. However, in certain cases the two may need to be reassociated:

- Someone has provided you with new contact information.
- An old contact has sent you a certificate to be associated with old contact information.

To associate a certificate with a contact:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Manage Trusted Identities**.
2. Choose a contact in the left-hand list ([Figure 33](#)).
3. Choose **Details**.
4. Choose **Associate Certificate** ([Figure 34](#)).

Figure 35 Contacts: Selecting certificates



5. Select a certificate from the list.
6. Choose **OK**.
7. Choose **OK**.

3.6.5 Changing a Trusted Identity's Certificate Association

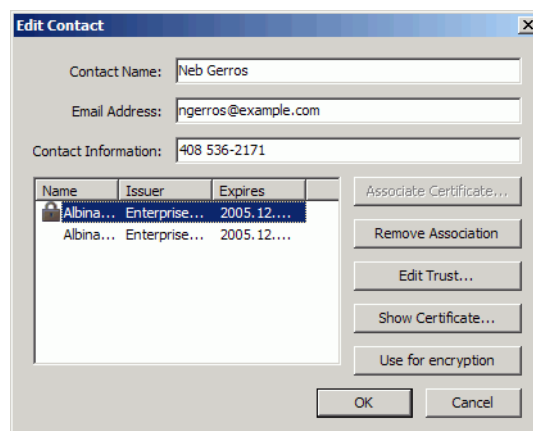
Contacts in the Trusted Identity Manager only have value when they are associated with certificates. Therefore, removing a certificate association only makes sense when it is being replaced by another certificate. For example, someone in your trusted identities list may have replaced a compromised or expired certificate with a new one. In this case, simply replace the old certificate association with a new one.

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Manage Trusted Identities**.
2. Choose a contact in the left-hand list (Figure 33).
3. Choose **Details**.
4. Choose a certificate from the list.
5. Choose **Remove Association** (Figure 36).
6. Choose a certificate from the list.

Note: The certificate list is populated with the currently associated certificate and any unassociated certificates for the current contact. In other words, the list does not display all of a contact's certificates, it displays only those that have no contact association.

7. Choose **Associate Certificate**.
8. Choose **OK**.

Figure 36 Edit Contact dialog



3.6.6 Deleting Contacts and Certificates

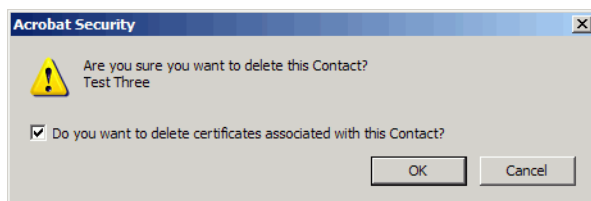
It is possible to delete contact information independently from its certificate. The most common scenarios for deleting trusted identity information include the following:

- You no longer share documents with someone and can delete all of their contact and certificate data.
- The trusted identity's contact information or certificate has changed and new data will be imported.

To delete a contact (and optionally a certificate):

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Manage Trusted Identities**.
2. Choose Contacts from the **Display** drop-down list.
3. Choose a contact in the left-hand list (Figure 33).
4. Choose **Delete**.
5. Choose whether or not to delete the certificates along with contact. Once a certificate is deleted, it can no longer be used to validate someone's signature or encrypt a document for them.
6. Choose **OK**.

Figure 37 Contacts: Deleting



Deleting a Certificate

To delete a certificate:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Manage Trusted Identities**.
2. Choose Certificates from the **Display** drop-down list.
3. Choose a certificate in the left-hand list (Figure 34).
4. Choose **Delete**.
5. Choose **OK**.

Acrobat's digital signature capabilities allow authors to set up a secure signing environment and create simple documents and complex forms with one or more fields. Document authors can design documents with multiple signature fields each with unique behavioral characteristics and appearances. A signed field can lock other fields so that signed data can't be changed, and authors can force certain signature fields to be a required part of a workflow. Attention to signature field design and configuration can help you make the document "do the right thing" when someone receives it as well as control what that person can and cannot do with it.

For more information, refer to the following:

- ["Best Practices for Signed Documents that will Change" on page 46](#)
- ["Setting up the Signing Environment" on page 46](#)
- ["Working with Signature Fields" on page 56](#)
- ["Authoring Signable Forms" on page 62](#)

4.1 Best Practices for Signed Documents that will Change

Some workflows require that someone enter form data, provide annotations and comments, or sign a document. When the form author has authorized or the application allows such changes, the changes are not flagged as problematic or warnings. In general, the goal should be to design documents and workflows so that both the signature status and document status are always valid.

Best practices for signed documents that will change vary by role:

- **Document author:** Form fields can be ordered, named, and associated with behaviors that limit changes in signing workflows.
- **First signer:** Use a certification signature for the first signature in a document and set **Permitted Changes after Certifying** as needed. The specified actions should not result in a warning triangle to appearing on signatures.
- **Signature validators and subsequent signers:** View the signed version of the document and look at the signature's status icon in the Signature's pane as well as the document status icon in the message bar. If there are any issues or problems, read the text. You may also wish to view the document the PDF signature report, view modifications, and so on.

4.2 Setting up the Signing Environment

A number of preferences control how your application, the document, and the signature will behave in signing workflows. These preferences tell the application where to look for Windows certificates, control signature appearances, enable the use of preview mode, and so on.

Tip: Participants in signing workflows (both document authors and signers) should review their application settings and configure their environment. Some preferences control authoring, some only have to do with signing, and some impact both.

Table 5 Signing environment preferences

Importance	Description	Section
Recommended	A number of preferences affect the signing workflow and resulting signature. All users should review these settings.	Setting Signing Preferences
Optional	Create default user information ahead of time to save time later.	Setting Identity Information
Optional	Use the default signature appearance or create your own.	Customizing Signature Appearances
Optional	To sign with a timestamp, configure a timestamp server.	Using Timestamps During Signing

4.2.1 Setting Signing Preferences

Preferences options vary by platform and application as follows:

- Choose one of the following:
 - Acrobat (Windows): **Edit > Preferences > Security**
 - Acrobat (Macintosh): **Acrobat > Preferences > Security**
 - Adobe Reader (Windows): **Edit > Preferences > Security**
 - Adobe Reader (Macintosh): **Adobe Reader > Preferences > Security**
- Set your preferences as described in the following sections:
 - [“Requiring Preview Mode” on page 47](#)
 - [“Changing the Default Signing Method” on page 48](#)
 - [“Embedding Signature Revocation Status” on page 49](#)
 - [“Allowing Signing Reason” on page 50](#)
 - [“Showing Location and Contact Details” on page 50](#)
 - [“Enabling Document Warning Review” on page 50](#)
 - [“Requiring Document Warning Review Prior to Signing” on page 51](#)
 - [“Enabling a Warnings Comment or Legal Attestation” on page 51](#)

4.2.1.1 Requiring Preview Mode

In general, everyone within a signing workflow needs to know that what they are viewing is actually what is signed or being signed. Preview mode automatically checks document integrity and generates a report that itemizes any content that could potentially prevent the signer from knowing what they are signing.

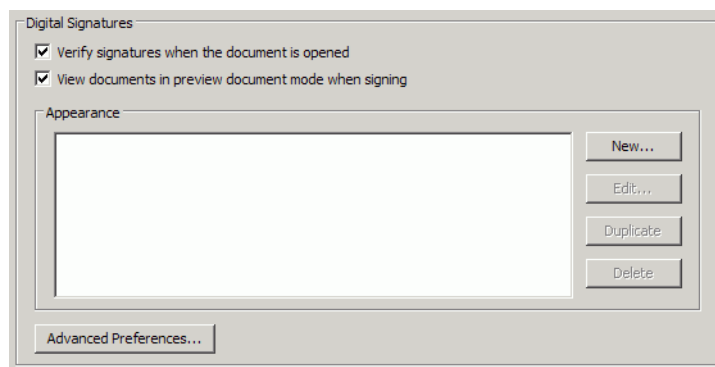
Preview mode provides several benefits:

- It checks the document for elements that may prevent a signer from seeing what they are signing.
- It suppresses document elements that may prevent a signer from knowing what they are signing.
- It generates a report about those elements. For details, see [Chapter 8, “Document Integrity and Preview Mode”](#).

To use preview mode automatically:

1. Choose one of the following:
 - Acrobat (Windows): **Edit > Preferences > Security**
 - Acrobat (Macintosh): **Acrobat > Preferences > Security**
 - Adobe Reader (Windows): **Edit > Preferences > Security**
 - Adobe Reader (Macintosh): **Adobe Reader > Preferences > Security**
2. Set **View documents in preview document mode when signing**.

Figure 38 Preview document mode preference



4.2.1.2 Changing the Default Signing Method

In some enterprise situations administrators may require a method other than Adobe Default Security. For example, non-Adobe plugins may be used in business environments that require support of biometrics, signature escrow, alternative methods of private key access, and so on. In those cases, administrators may preconfigure Acrobat to use an alternate plugin or provide user training on how to choose the right one.

Third party plugins include:

- **Entrust® plug-in for Acrobat 4 and 6:** This plugin interfaces to the Entrust Entelligence desktop application and provides the same functionality that is provided by Adobe's plugin. Businesses that use Entrust for PKI deployment may require the Entrust plug-in.
- **SignCube® plug-in for Acrobat 7:** The SignCube plugin is used to create signatures recognized as valid under the German Digital Signature Law.
- **CIC:** The Communication Intelligence Corporation® Plugin (CIC) is used by some banks and insurance companies to provide an electronic version of handwritten signatures. This plugin limits users' ability to use encryption.

To change the default signing method:

1. Choose one of the following:
 - Acrobat (Windows): **Edit > Preferences > Security**
 - Acrobat (Macintosh): **Acrobat > Preferences > Security**
2. Choose **Advanced Preferences**.
3. Choose the Creation tab (Figure 39).

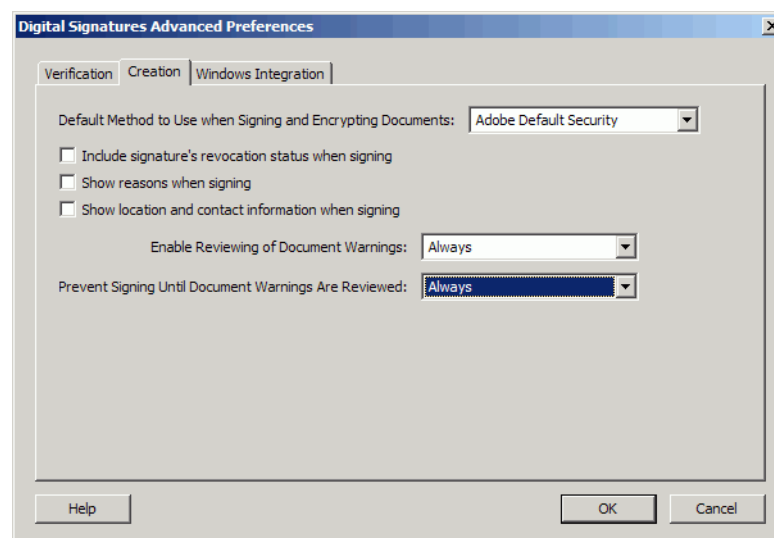
4. Under **When Verifying**, check one of the last two radio buttons so that the **Default Method for Verifying Signatures** drop down list becomes active.
5. Select a signing method.

Note: Do not change the signing method unless instructed to do so by your administrator.

4.2.1.3 Embedding Signature Revocation Status

1. Choose one of the following:
 - Acrobat (Windows): **Edit > Preferences > Security**
 - Acrobat (Macintosh): **Acrobat > Preferences > Security**
 - Adobe Reader (Windows): **Edit > Preferences > Security**
 - Adobe Reader (Macintosh): **Adobe Reader > Preferences > Security**
2. Choose **Advanced Preferences**.
3. Choose the Creation tab (Figure 39).

Figure 39 Signature creation preferences



4. (Recommended) **Set Include signature's revocation status when signing.**

Embedding the signing certificate's revocation status in a document allows recipients to validate certificates (signatures) while offline and speeds up the revocation checking process. Moreover, if a certificate is revoked or expired at some time after signing, embedded revocation information enables the application to determine if a certificate was valid at the time of signing so that the signature status will remain valid.

Note: Revocation checking occurs immediately after signing as well as during signature validation. If the revocation status is not embedded in a signature, the application looks in the locally stored certificate revocation list cache. If it is not there, the application goes online to complete the check.

4.2.1.4 Allowing Signing Reason

Turning this option on results in a **Reasons** field appearing in the signing dialog. The signer can then choose a default reason such as “I have reviewed this document” or create a new one.

1. Choose one of the following:
 - Acrobat (Windows): **Edit > Preferences > Security**
 - Acrobat (Macintosh): **Acrobat > Preferences > Security**
 - Adobe Reader (Windows): **Edit > Preferences > Security**
 - Adobe Reader (Macintosh): **Adobe Reader > Preferences > Security**
2. Choose **Advanced Preferences**.
3. Choose the Creation tab (Figure 39).
4. Set **Show reasons when signing**.

4.2.1.5 Showing Location and Contact Details

When this option is turned on, the **Location** and **Contact Info** fields appear in the signing dialog.

1. Choose one of the following:
 - Acrobat (Windows): **Edit > Preferences > Security**
 - Acrobat (Macintosh): **Acrobat > Preferences > Security**
 - Adobe Reader (Windows): **Edit > Preferences > Security**
 - Adobe Reader (Macintosh): **Adobe Reader > Preferences > Security**
2. Choose **Advanced Preferences**.
3. Choose the Creation tab (Figure 39).
4. Set **Show location and contact information when signing**.

4.2.1.6 Enabling Document Warning Review

Enabling document warning review allows signers to check document integrity prior to signing. The document can be analyzed to determine if it contains any content that could adversely impact the integrity of the signing process. For example, a document could contain JavaScript that could change a data field before or after a signature is applied. Setting an option here affects what appears in the **Prevent Signing Until Document Warnings Are Reviewed** drop-down list.

Note: **Enable Reviewing of Document Warnings** and **Prevent Signing Until Document Warnings Are Reviewed** settings function in tandem and should be set together. Setting both these options to *Always* results in the highest degree of assurance that the signing process is not adversely impacted by malicious content.

1. Choose one of the following:
 - Acrobat (Windows): **Edit > Preferences > Security**
 - Acrobat (Macintosh): **Acrobat > Preferences > Security**
 - Adobe Reader (Windows): **Edit > Preferences > Security**

- Adobe Reader (Macintosh): **Adobe Reader > Preferences > Security**
2. Choose **Advanced Preferences**.
 3. Choose the Creation tab (Figure 39).
 4. Set **Enable Reviewing of Document Warnings**. Select from the following:
 - **Never**: Turns off document warning review. No **Review** button appears in the signing dialog.
 - **When certifying a document**: The **Review** button appears in the signing dialog only when a certification signature is being applied. This option allows the signer to add a legal attestation.
 - **Always**: The **Review** button always appears in the signing dialog. This option allows the signer to add a legal attestation for certification signatures.

4.2.1.7 Requiring Document Warning Review Prior to Signing

If document warning reviews are critical to your signing workflow, you can require them.

Note: **Enable Reviewing of Document Warnings** and **Prevent Signing Until Document Warnings Are Reviewed** settings function in tandem and should be set together. Setting both these options to *Always* results in the highest degree of assurance that the signing process is not adversely impacted by malicious content.

1. Choose one of the following:
 - Acrobat (Windows): **Edit > Preferences > Security**
 - Acrobat (Macintosh): **Acrobat > Preferences > Security**
 - Adobe Reader (Windows): **Edit > Preferences > Security**
 - Adobe Reader (Macintosh): **Adobe Reader > Preferences > Security**
2. Set **Prevent Signing Until Document Warnings Are Reviewed**. Select from the following:
 - **Never**: Signing can continue without a document warning review.
 - **When certifying a document**: Signers must choose **Review** to apply a certification signature.
 - **Always**: Signers must always choose **Review** when signing.

4.2.1.8 Enabling a Warnings Comment or Legal Attestation

For certified documents that contain multimedia, comments, or other dynamic content, it is often beneficial to add a warnings comment or legal attestation that states that the content is OK and permitted by the author. If document warnings are enabled, then the signer can review the warnings and either choose from Acrobat's default comment "I have included this content to make the document more dynamic," or create a custom comment.

To enable warnings comments on certified documents:

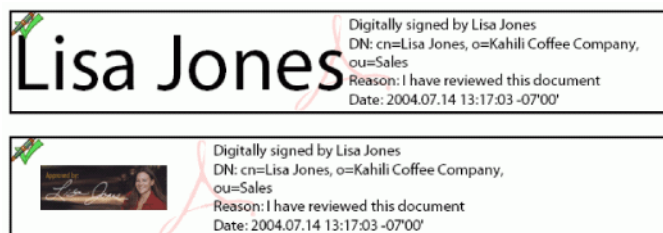
1. Set **Enable Reviewing of Document Warnings** to **When certifying a document** or **Always** as described in ["Enabling Document Warning Review" on page 50](#).

4.2.2 Customizing Signature Appearances

The signing process allows you to select from a list of signature appearances. Acrobat provides a default appearance, but you can create one or more custom appearances and store them for later use. An appearance consists of three main components, and each can be separately customized:

- **Signature:** The text or graphic that identifies the signer on the left hand side of the appearance.
- **Watermark or background:** A background image or text that is automatically applied to each signature. By default, this is the Adobe PDF logo.
- **Signature text and details:** Signature data that the signer want to include in the appearance.

Figure 40 Custom signature appearance



Tip: To learn how to control appearances programmatically, developers should refer to [Digital Signature Appearances](#).

4.2.2.1 Creating a Custom Signature

Acrobat creates a default signature from the signer's name. However, a signature can be any graphic such as a scanned signature, text, or a combination of the two ([Figure 40](#)). Make the background transparent if watermark should be visible in the underlying layer.

To create a signature:

1. Create a graphic from your scanned signature or from some other image.
2. Make the background transparent if desired.
3. Save the file to any location.

4.2.2.2 Creating a Custom Watermark or Background

A watermark is a partially transparent graphic or logo that appears "behind" a signature. By default, the watermark is the Adobe PDF logo. Line (vector) art that is simple and unobtrusive often works best.

1. Import a logo or create a new one in a program such as Adobe Illustrator.
2. Set a low transparency level and flatten the transparency:
 1. Select all and group the objects if there is more than one.
 2. Choose **Window > Transparency** and slide the transparency slider to some low value such as 20%.
 3. Choose **Object > Flatten Transparency**. Leaving the Raster/Vector balance at 100%.

4. Save the file to a PDF file.
5. Open the PDF file in Acrobat.
6. Crop the page and remove white space.

Note: The method varies across product versions. For example, for 8.x, choose **Document > Crop Page** and check **Remove White Margins**.

7. Save the file as SignatureLogo.pdf in:
 - **Windows:** C:\Documents and Settings\<user>\Application Data\Adobe\Acrobat\<version>\Security.
 - **Macintosh:** \Users\<user name>\Library\Application Support\Adobe\Acrobat\<version>\Security

4.2.2.3 Creating a Custom Signature Appearance

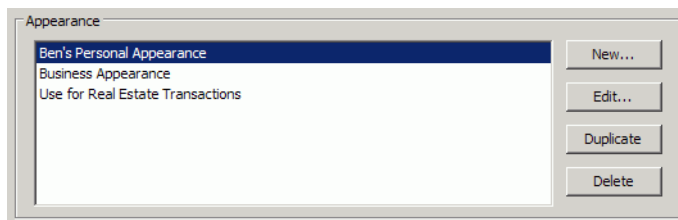
Users generally customize one or more signature appearances and store them for later use. Available signatures are listed in the Appearance panel ([Figure 41](#)).

Note: If you have created a watermark file as described in [“Creating a Custom Watermark or Background” on page 52](#), the watermark should automatically appear in all of your signature appearances.

To create a new signature appearance:

1. Choose **Edit > Preferences** (Windows) or **Acrobat > Preferences** (Macintosh).
2. Choose **Security** in the left-hand list.
3. In the Appearance panel, choose **New**.

Figure 41 Signature appearance: New button



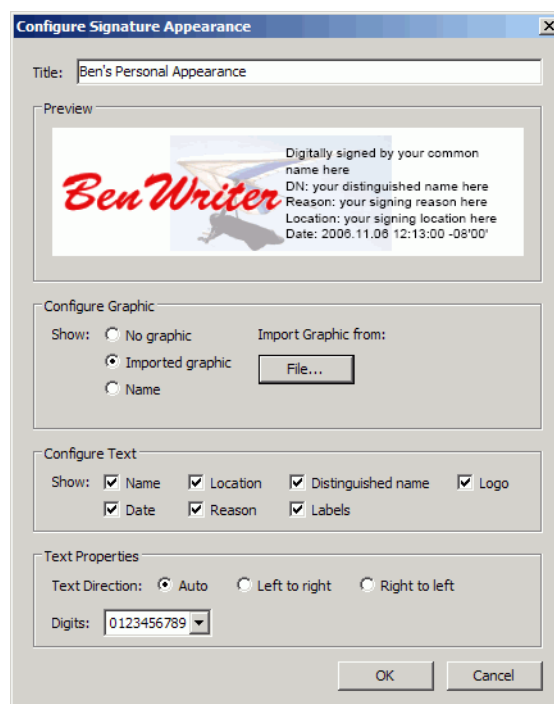
4. Configure the signature appearance options:
 - **Title:** Any arbitrary title used to identify the appearance.
 - Set the graphic options in the Graphic panel
 - **No graphic:** No graphic is used.
 - **Imported graphic:** Choose **File > Browse**, select a file and choose **OK**.
 - **Name:** Your text name will appear instead of a graphic. The name is extracted from the signing certificate.

Note: By default, the signature watermark is the Adobe PDF logo but it can be customized. To avoid obscuring a background, use line art with a transparent background.

- Set the text options in the Configure Text panel
 - **Name:** The name associated with the certificate.
 - **Date:** The date signed.
- **Note:** Signature appearances can only display local (computer) time, and it will likely differ from that in the Date/Time tab on the Signature Properties dialog when a timestamp server is used.
- **Location:** The location associated with the identity configured in Acrobat.
- **Reason:** The reason for signing.
- **Distinguished name:** A name with details such as country, organization, organizational unit, and so on.
- **Labels:** A label for each of the items above. For example, *Reason*.
- **Logo:** The logo or graphic used as a background watermark.
- Set the text options in the Text Properties panel
 - **Text Direction:** Choose a direction appropriate for the signer's language.
 - **Digits:** If languages are installed that use digits other than 1234567890, the drop-down list will be populated with alternate choices. Choose a digit set appropriate for the signer's language.

5. Choose **OK**.

Figure 42 Signature appearance: Configuration



4.2.2.4 Editing or Deleting a Signature Appearance

Existing signature appearances can be edited at any time.

To edit a signature appearance:

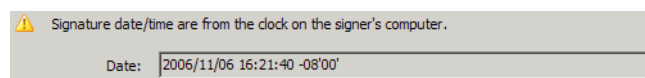
1. Choose **Edit > Preferences** (Windows) or **Acrobat > Preferences** (Macintosh).
2. Choose **Security** in the left-hand list.
3. Highlight an appearance in the Appearance panel.
4. Choose **Edit** or **Delete**.
5. Edit the appearance. For details, see [“Creating a Custom Signature Appearance” on page 53](#).
6. Choose **OK**.

4.2.3 Using Timestamps During Signing

Signature times tell you that a document and signature existed prior to the indicated time. All signatures are associated with the signer machine's local time, but they may also include a timestamp time provided by a timestamp server if one is configured. Because a user can set that time forward or back, a local time is less reliable than a timestamp time. Local times are labelled as such in the Date/Time and Summary tabs of the Signature Property dialog ([Figure 43](#)).

Note: Because signature appearances only display local time, the appearance time will be different from the timestamp time shown in the Date/Time tab of the Signature Properties dialog.

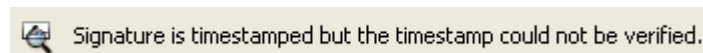
Figure 43 Timestamps: Local, machine time



Like signatures, timestamps are provided by someone (a timestamp authority) who uses certificates to confirm their identity. Before you can validate a timestamp, you must explicitly trust the timestamp authority's certificate. Timestamp certificate status appears in the Date/Time and Summary tabs of the Signature Property dialog:

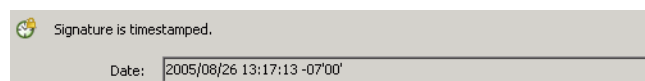
- Untrusted timestamp certificates appear as follows:

Figure 44 Timestamps: Untrusted stamp



- Trusted timestamps that have been added to the Trusted Identities list and have been explicitly trusted for signing appear as follows:

Figure 45 Timestamps: Trusted stamp



Timestamps are usually provided by third-party timestamp authorities such as GeoTrust. Because timestamp authorities may charge for their services, Acrobat does not automatically set a default timestamp server if multiple servers are listed. Users must manually specify which timestamp server to use as the default.

Configuring Acrobat to use a Timestamp Server

To use a timestamp when you sign, configure your application to use a timestamp server, set it as the default, and set trust the certificate of the timestamp authority. The timestamp server is always used if a default timestamp server is specified.

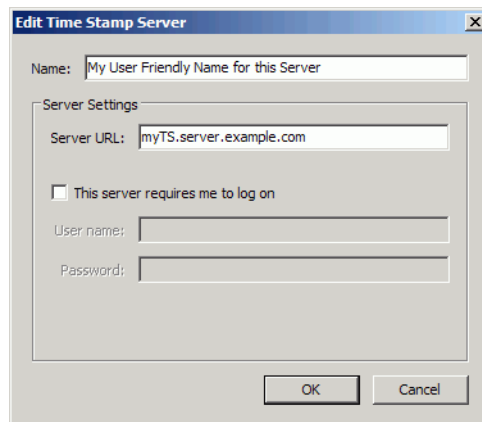
To manually set up a timestamp server:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Choose **Timestamp Servers**.
3. Choose **New**.

In most cases, administrators preconfigure end user machines or provide the server information in an FDF file. If you have an FDF file, see ["Importing Timestamp Server Settings" on page 169](#).

4. Enter the server settings:
 - **Name:** The server name.
 - **Server URL:** The server URL.
 - **Username:** The login username if required.
 - **Password:** The login password if required.
5. Choose **OK**.
6. Make this server or some other server the default by choosing **Set Default**. Timestamping cannot occur unless a default server selected.

Figure 46 Timestamps: Entering server details



4.3 Working with Signature Fields

Signature fields are a type of form field, and both Acrobat and Adobe Reader ignore whether they are authored with Forms Designer or Acrobat. Digital signatures behave uniformly irrespective of the authoring mechanism.

For details about customizing one or more fields, see the following:

- [Specifying General Field Properties](#)
- [Customizing Field Appearances](#)
- [Changing the Default Field Appearance](#)
- [Cut, Copy, and Paste Signature Fields](#)
- [Arranging Signature Fields](#)
- [Creating Multiple Copies of a Signature Field](#)
- [Authoring a Document with Multiple Fields](#)
- [Locking Fields Automatically After Signing](#)
- [Unlocking a Field Locked by a Signature](#)
- [Making a Field a Required Part of a Workflow](#)
- [Specifying a Post-Signing Action](#)

4.3.1 Creating a Blank Signature Field

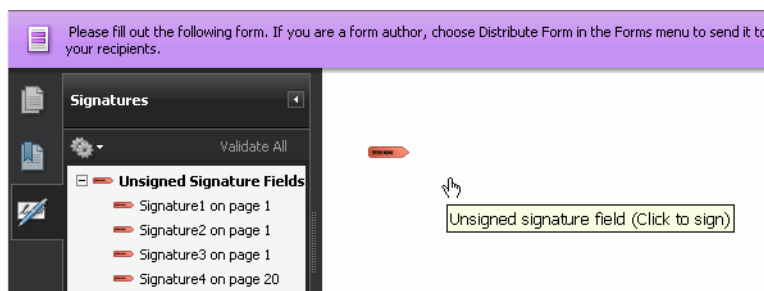
Signatures and related information are stored in a signature field embedded on the page. A signature field is an Acrobat form field. Signature fields are automatically created at the time of signing, but it is also possible to create empty signature fields for later signing.

To create a signature field:

1. Choose **Forms > Add or Edit Fields**.
2. Select the **Add New Field** button.
3. Choose **Digital Signature**.
4. Click and drag where the field should appear. The Digital Signature Properties dialog appears.
5. For simple signature fields, choose **Close**.

By default, field names are numbered sequentially starting with “Signature1” and contain the default tooltip “Unsigned signature field (click to sign).”

Figure 47 Signature field: Default appearance



6. If you would like to set field properties, choose the **Show All Properties** link in the popup box (for details, see [“Specifying General Field Properties” on page 58](#)); otherwise, exit the forms editor.

4.3.2 Specifying General Field Properties

A signature field's general properties include name, tooltip, display behavior, and so on. For example, fields are numbered sequentially and are associated with a generic tooltip. However, the field can be given a unique name, provided with tooltip instructions for an eventual signer, and configured to display only in the Signatures tab and not in the document.

Note: You cannot edit these properties during signing workflows. An author must create a blank signature field and edit the properties before initiating the signing process. Moreover, invisible field properties cannot be edited.

To change a field's general properties.

1. Create a new field.

Note: For existing fields, place them field in edit mode by selecting **Forms > Add or Edit Fields** and then double click on them OR right click and choose **Properties**.

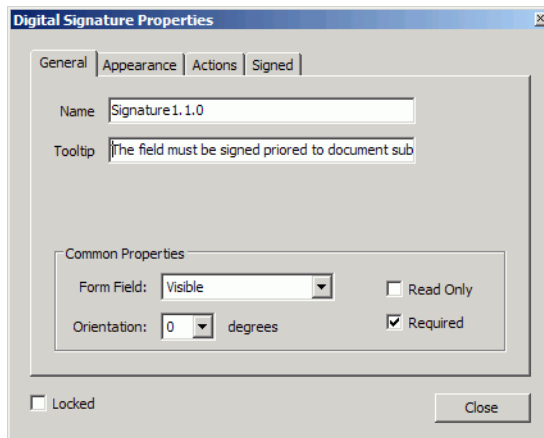
2. Display the General tab.

3. Configure the options:

- **Name:** Any arbitrary name.
- **Tooltip:** Any arbitrary text. It is required to make the document accessible to the visually impaired.
- **Form Field:** Set the field display properties.
 - **Visible:** The field appears in the document, the Signatures tab, and prints.
 - **Hidden:** The field only appears on the Signatures tab and doesn't print.
 - **Visible but doesn't print:** The field appears in the document and Signatures tab but doesn't print.
 - **Hidden but printable:** The field only appears on the Signatures tab and does print.
- **Orientation:** The field content (signature) orientation AFTER signing. The field does not change.
- **Read Only:** Prevents changes to the field. Selecting this option prevents signing.
- **Required:** Sets a flag that can be checked by other actions and processes that are dependent on the signature. For details, see ["Making a Field a Required Part of a Workflow" on page 63](#).

4. Edit the properties on the other tabs or choose **Close**.

Figure 48 Signature field: General properties



4.3.3 Customizing Field Appearances

Field border properties, fill color, fonts, and so on can be individually specified. These properties are NOT editable during signing workflows. An author must create a blank signature field and edit the properties before initiating the signing process. Invisible field properties cannot be edited.

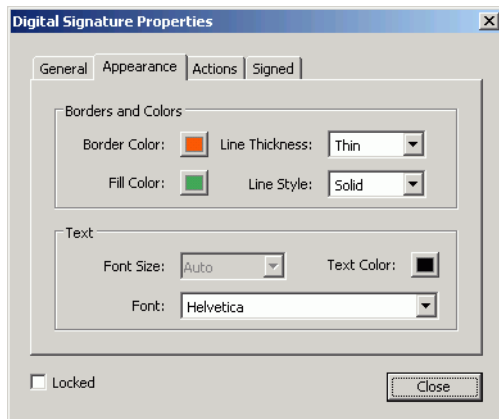
To change a signature field's appearance:

1. Create a new field.

Note: For existing fields, place them field in edit mode by selecting **Forms > Add or Edit Fields** and then double click on them OR right click and choose **Properties**.

2. Display the Appearance tab.
3. Configure the appearance options.
4. Edit the properties on the other tabs or choose **Close**.

Figure 49 Signature field: Appearance properties



4.3.4 Changing the Default Field Appearance

The default appearance of a blank signature field is a light blue box with no borders that performs no action on signing. These defaults can be changed globally so that all future signature fields will have a custom appearance and action.

To change signature field defaults:

1. Only the attributes on the Appearance and Actions tab can be set as defaults for future fields. Customize a field as described in the following.
 - [Customizing Field Appearances](#)
 - [Specifying a Post-Signing Action](#)
2. Choose **Close**.
3. Right click on the field.
4. Choose **Use Current Properties as New Defaults**.

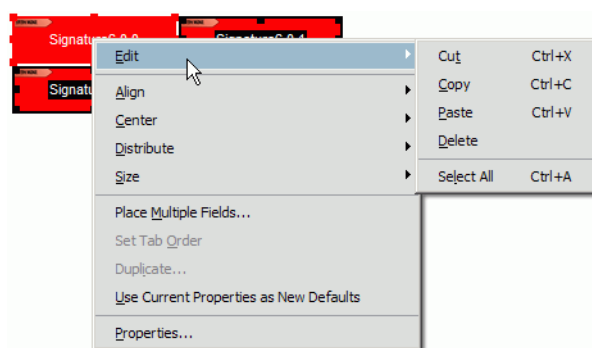
4.3.5 Cut, Copy, and Paste Signature Fields

The forms field context menu provides a number of editing items, including options for cutting, copying, pasting, and deleting.

To perform an edit action on a field:

1. Place the field in edit mode by selecting **Forms > Add or Edit Fields**.
2. Right click on the field.
3. Choose **Edit**.
4. Choose **Cut, Copy, Paste, or Delete** (Figure 50).

Figure 50 Signature field: Edit options



4.3.6 Arranging Signature Fields

While you can drag and drop fields anywhere, the field context menu provides a number of options for arranging multiple fields such as aligning, centering, and distributing fields.

To arrange multiple fields:

1. Place the fields in edit mode by selecting **Forms > Add or Edit Fields**.
2. Drag a rectangle around the fields to arrange.
3. Right click.
4. Choose **Align**, **Center**, or **Distribute** and use the submenus to arrange the fields (Figure 50).

4.3.7 Creating Multiple Copies of a Signature Field

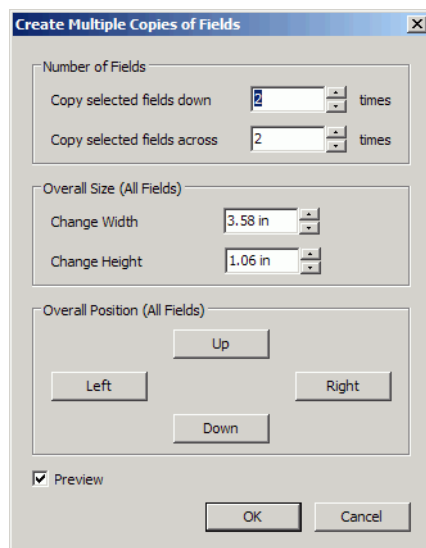
Once a field is configured, multiple copies of the field can be placed on the same page.

To create multiple copies of a field:

1. Place the field in edit mode by selecting **Forms > Add or Edit Fields**.
2. Right click on the field.
3. Choose **Create Multiple Copies** (Figure 50).
4. Configure the copy options, including:
 - The number of fields down and across.
 - The overall field size.
 - The overall position.
5. Choose **OK**.

Tip: Acrobat automatically names the fields by numbering them. Providing unique and intuitive names helps signers and other document recipients navigate and interact with the document.

Figure 51 Signature field: Multiple copy options



4.4 Authoring Signable Forms

Many documents that require signatures are forms. Some forms may have multiple signatures fields, with different signers providing data in certain other form fields. In such cases, document design, field layout, and even field appearance may contribute to the ease with which the form can be integrated into an efficient business process.

For example, it is often useful to lock the form fields associated with a particular signature field once it is signed. This eliminates the need to examine the signed document version to see if the value of a field was changed between that signed version and the current version. For more information, see the following:

- [“Authoring a Document with Multiple Fields” on page 62](#)
- [“Locking Fields Automatically After Signing” on page 62](#)
- [“Making a Field a Required Part of a Workflow” on page 63](#)
- [“Specifying a Post-Signing Action” on page 64](#)
- [“Unlocking a Field Locked by a Signature” on page 66](#)

4.4.1 Authoring a Document with Multiple Fields

Documents commonly have multiple form fields, and one or more signature fields are often used to verify or approve the data in preceding fields. In these cases, proper document layout and field design may be a critical aspect of usability. When designing a complex form for signing, consider using the following field properties:

- **Layout:** Design the form so that form data precedes a signature. If there is more than one signature field, make sure end users can understand which signature fields are associated with specific data.
- **Appearance:** Signature fields can look similar to other form fields, but it may be desirable to customize their appearance so they can be more readily distinguished. For details, see [“Customizing Field Appearances” on page 59](#)
- **Names and tooltips:** Intuitive field names and tooltips facilitate authoring and signing in the following ways:
 - Help the author choose which fields should be read only in the Signed tab of the Digital Signature Properties dialog as well as what field to call when JavaScript is used to customize a document.
 - Help signers find fields and understand how to use the form. For details, see [“Specifying General Field Properties” on page 58](#).
 - Make it easier for signature validators to identify which fields have changed since the names may be used in the Signature pane and elsewhere.
- **Locking behavior:** Consider which fields should become read-only after signing. Locking certain fields helps prevent document changes that could cause a signature to become invalid. For details, see [“Locking Fields Automatically After Signing” on page 62](#).

4.4.2 Locking Fields Automatically After Signing

Form authors can designate which form fields should be locked after any other field is signed. Both signed and unsigned signature fields can be configured to become read-only after they are signed. By setting post-signing, field locking properties, authors can prevent data changes to any combination of form or signature fields. Two common use cases for automatic locking include:

- Preventing users other than the document author from clearing or re-signing a field.

- Preventing users from changing form data after the document has been signed.

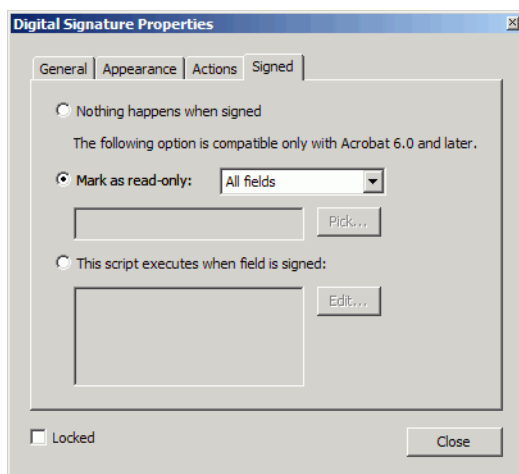
To automatically lock one or more fields after signing:

1. Create a new signature field.

Note: For existing fields, place them field in edit mode by selecting **Forms > Add or Edit Fields** and then double click on them OR right click and choose **Properties**.

2. Display the Signed tab.

Figure 52 Signature field: Signing properties



3. Choose **Mark as read only**.

When a field with field locking is signed, both a normal document signature and an object hash of the locked fields are produced and included in the document. When the signature is validated, the viewing application validates the bytes of the PDF file AND compares the object hash in the signature to the object hash from the objects in memory. This allows the application to detect prohibited, changes.

4. Use the drop-down list to select from the following:

- **All fields:** All signature fields will be read only after signing.
- **All fields except these:** All signature fields except those specified will be read only after signing. Choose **Pick** and select the field to exclude.
- **Just these fields:** Only the specified signature fields will be read only after signing. Choose **Pick** and select the field to include.

5. Choose **Close**.

4.4.3 Making a Field a Required Part of a Workflow

Certain workflows may require a signature. For example, after a signature field is signed, form fields may be prepopulated or additional fields may appear. It is also common for form designers to require signing before the document can be emailed or submitted to a server for processing.

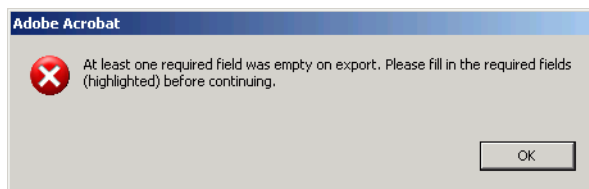
To require a signature:

1. Place the field in edit mode by selecting **Forms > Add or Edit Fields**.

2. Right click on the field and choose **Properties**.
3. Check **Required** on the General tab.
4. Choose **Close**.

Users can still open, close, save, and send the document without any indication that the field is required until the document author sets up a check for the required flag. For example, the check could be as simple as emailing the document. In this case, the author would add an action to the button to submit the document and configure a URL. If the document recipient clicks on the field and then cancels the signing process, an alert will appear. Server-side and other JavaScript checks are also possible.

Figure 53 Required field not signed alert



4.4.4 Specifying a Post-Signing Action

JavaScript actions can be associated with a signature field so that an action occurs whenever the user interacts with the field in some predefined way. However, documents are usually signed to protect, guarantee, and or attest to the signed content. Signers usually want to know that the document they are seeing is the document they are signing, and document recipients usually need to know that the document they are viewing is the same as the document that was signed. For this reason, adding actions to a signature field is inadvisable. Field actions change the underlying bytes of a PDF and could adversely affect document security as well as content integrity.

Caution: Using this feature is NOT recommended since such actions are contrary to the secure and trusted nature of most signing workflows. Adding actions will result in a legal warning about the legal integrity of the document.

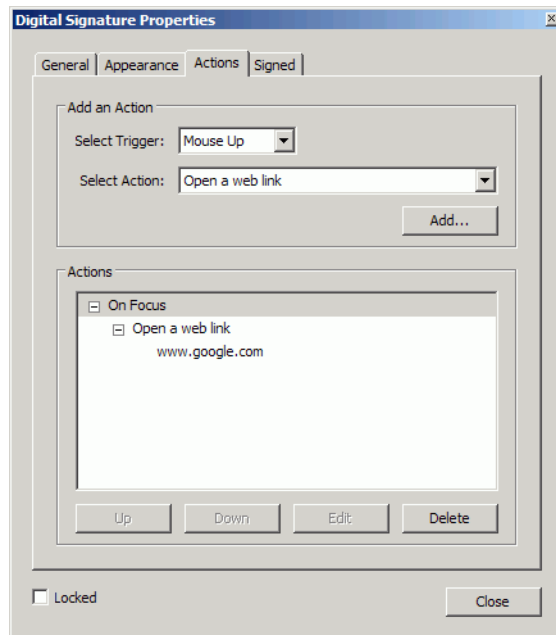
To associate an action with a field.

1. Create a new field.

Note: For existing fields, place the form in edit mode by selecting **Forms > Add or Edit Fields** and then double click on them OR right click and choose **Properties**.

2. Display the Action tab.

Figure 54 Signature field: Action properties



3. Configure the options:
 - **Select Trigger:** Choose a type of action.
 - **Mouse Up:** The user clicks on the field and releases.
 - **Mouse Down:** The user clicks on the field.
 - **Mouse Enter:** The cursor enters the field.
 - **Mouse Exit:** The cursor exits the field.
 - **On Focus:** The user hovers over or tabs to the field.
 - **On Blur:** The user stops hovering over or tabs away from the field.
 - **Select Action:** See [Table 6](#).
4. Choose **Add**.
5. Follow the action instructions that appear in the action dialog.
6. Optional: Move actions **Up**, **Down**, **Edit**, or **Delete** actions as necessary.
7. Choose **Close**.

Table 6 Actions that can be associated with a signature field

Action	Description
Execute a Menu Item	Executes a specified menu command as the action.
Go to 3D View	Changed the view to the 3D view specified by the form author.
Go to a Page View	Jumps to the specified destination in the current document or in another document.
Import Form Data	Brings in form data from another file, and places it in the active form.

Table 6 Actions that can be associated with a signature field

Action	Description
Open a File	Launches and opens a file. If you are distributing a PDF file with a link to a non-PDF file, the reader needs the native application of the non-PDF file to open it successfully. (You may need to add opening preferences for the target file.)
Open a Web Link	Jumps to the specified destination on the Internet. You can use http, ftp, and mailto protocols to define your link.
Play a Sound	Plays the specified sound file. The sound is embedded into the PDF document in a cross-platform format that plays in Windows and Macintosh.
Play Media (Acrobat 5 Compatible)	Plays the specified QuickTime or AVI movie that was created as Acrobat 5-compatible. There must already be a link to the movie in the PDF document for you to be able to select it. (See Adding movie clips.)
Play Media (Acrobat 6 Compatible)	Plays a specified movie that was created as Acrobat 6-compatible. There must already be a link to the movie in the PDF document for you to be able to select it. (See Adding movie clips.)
Read an Article	Follows an article thread in the active document or in another PDF document.
Reset a Form	Clears previously entered data in a form. You can control the fields that are reset with the Select Fields dialog box.
Run a JavaScript	Runs the specified JavaScript.
Set Layer Visibility	Determines which layer settings are active. Before you add this action, specify the appropriate layer settings.
Show/Hide a Field	Toggles between showing and hiding a field. This option is especially useful in form fields. For example, if you want an object to pop up whenever the pointer is over a button, you can set an action that shows a field on the Mouse Enter trigger and hides a field on Mouse Exit.
Submit a Form	Sends the form data to the specified URL.

4.4.5 Unlocking a Field Locked by a Signature

When a signature field's properties specify that signing will automatically lock other fields, those fields cannot be edited until they are unlocked. Since it was a signature that locked the fields in the first place, unlocking the fields simply involves clearing the signature.

Tip: Unlocking a field is the same thing as clearing the signature field. Only the signer clear the signed field.

To unlock a field:

1. Right click on the signature field.
2. Choose **Clear Signature**.
Other fields can now be edited.

Acrobat's seed value feature helps authors control document behavior once it has been routed to the signer. Seed values can be used to embed certificate requirements and other instructions in signature fields. When a signers signs a custom, "seeded" field, the author-specified behaviors are automatically invoked and enforced.

A seed value specifies an attribute and attribute value. The author can make the seed value a preference or a requirement. For example, You can use seed values to limit a user's choices when signing a particular signature field. For details about what seed values can be used to control, see the following:

- [Forcing a Certification Signature](#)
- [Giving Signers the Option to Lock a Document](#)
- [Forcing Signers to Use a Specific Signature Appearance](#)
- [Adding Custom Signing Reasons](#)
- [Specifying Timestamps for Signing](#)
- [Specifying Alternate Signature Handlers and Formats](#)
- [Specifying a Signature Hash Algorithm](#)
- [Embedding Revocation Information in a Signature](#)
- [Specifying Certificate Properties for Signing](#)
 - [Specifying Signing Certificates Origin](#)
 - [Specifying Certificates by Key Usage](#)
 - [Specifying Certificates by Policy](#)
 - [Specifying a URL When a Valid Certificate is not Found](#)
- [Custom Workflows and Beyond](#)

5.1 Seed Value Basics

To set seed values for Acrobat forms, JavaScript calls must be used because no direct user interface is provided. To set seed values for LiveCycle Forms, the Adobe LiveCycle Designer user interface can be used to set seed values in the signature field Properties panel.

The JavaScript function `signatureSetSeedValue` sets properties that are used when signing signature fields. The properties are stored in the signature field and are not altered when the field is signed, the signature is cleared, or when `resetForm` is called. This method (and JavaScript generally) can be executed with a batch process, by dropping the script in Acrobat's JavaScript subdirectory, menu events, Acrobat's JavaScript debugger, and other methods.

When setting seed values, keep in mind the following:

- Seed values should not be set on signed documents and cannot be set on certified documents after the document is certified. They are primarily used to configure fields on documents that are not yet signed.
- Setting a seed value often causes Acrobat to not display or use its default settings. For example, default reasons are stored in a registry list, and specifying signing reasons with a seed value overrides that list.
- Seed value properties include those listed in [Table 8](#). Note that `certspec` and `timeStampspec` are objects that have multiple properties.

5.1.1 Changes Across Releases

Each Acrobat release results in support for additional seed values as shown in [Table 7](#).

Table 7 Seed values: Changes across releases

Seed value	First support for seed value	6	7	8	9
certspec	Specifies that certain certificates must be used for a particular signature field. 6.0-7.x: Supports subject, issuer, and oid. 8.x: Adds support for subjectDN, issuerDN, keyUsage, url, and urlType	X	X	X	X
filter	The language-independent name of the security handler to be used when signing.	X	X	X	X
flags	A set of bit flags controlling which properties are required. 6.0-7.x: 1: filter, 2: subFilter, 4: version, and 8: reasons. 8.0: 16: legalAttestations, 32: shouldAddRevInfo, and 64: digestMethod.	X	X	X	X
legalAttestations	A list of legal attestations that the user can use when creating an MDP (certification) signature.	X	X	X	X
mdp	Can be used to force a certification signature as well as to control permitted document changes.	X	X	X	X
reasons	A list of reasons that the user is allowed to use when signing. 8.0: Supports disabling signing reasons.	X	X	X	X
subFilter	An array of acceptable signature formats.	X	X	X	X
timeStampspec	Specifies a timestamp server using the <code>url</code> and <code>flags</code> properties.	X	X	X	X
version	The signature handler version to be used to sign the signature field. Valid values are 1 and 2. 8.0: Must be set to 2 if this seed value object contains Acrobat 8-specific content marked as required.	X	X	X	X
digestMethod	The algorithm used to created the message digest. 6.0-7.x: MD5, SHA1. 8.0: Adds support for SHA256, SHA384, SHA512, and RIPEMD160. Note: SHA256, SHA384, and SHA512 are not supported in Windows CAPI until Acrobat 9.1, and not supported in Windows at all prior to XP-SP3.	X	X	X	X
shouldAddRevInfo	Controls how the application does certificate and chain revocation checking.			X	X
lockDocument	Allows the author to add a Lock Document checkbox to the signing dialog so a signer can lock the document at the time of signing.				X
appearanceFilter	A text string naming the appearance required to be used when signing the signature field.				X

5.1.2 Supported Seed Values

Note: The examples in this document demonstrate the simplest case. For more information, refer to the *Acrobat JavaScript Scripting Guide* and *JavaScript for Acrobat API Reference*.

Table 8 Seed values: Object properties and descriptions

Property	Type	Description
certspec	object	A seed value CertificateSpecifier Object. For details, see “Specifying Certificate Properties for Signing” on page 80 .
digestMethod	array of strings	(Acrobat 8.0) An array of acceptable digest methods to use while signing. These are only applicable if the digital ID contains RSA public and private keys. If they contain DSA public/private keys, then the value is always SHA1. Valid values include: MD5, SHA1 (default), SHA256, SHA384, SHA512, and RIPEMD160.
filter	string	The language-independent name of the signature handler to be used when signing.
flags	number	<p>A set of bit flags controlling which of the following properties are required. The value is the logical OR of the following values, which are set if the corresponding property is required:</p> <ul style="list-style-type: none"> 1: filter 2: subFilter 4: version 8: reasons 16: legalAttestations (Acrobat 8.0) 32: shouldAddRevInfo (Acrobat 8.0) 64: digestMethod (Acrobat 8.0) 128: lockDocument 256: appearanceFilter <p>Usage: 1 specifies filter, 3 specifies filter and sub-filter, and 11 specifies filter, sub-filter, and reasons. If this field is not present, all properties are optional.</p> <p>For more details, refer to the <i>PDF Reference</i>.</p>
version	real	<p>See version (above). (Optional) The minimum required capability of the signature field seed value dictionary parser. A value of 1 specifies that the parser must be able to recognize all seed value dictionary entries specified in PDF 1.5. A value of 2 specifies that it must be able to recognize all seed value dictionary entries specified in PDF 1.7 and earlier. A value of 3 specifies that it must be able to recognize all seed value dictionary entries specified in PDF1.7-ADBE-3 and earlier.</p> <p>The Ff (flags above) entry indicates whether this is a required constraint.</p> <p>For more details, refer to the <i>PDF Reference</i>.</p>
legalAttestations	array of strings	(Acrobat 7.0) A list of legal attestations that the user can use when creating an MDP (certified) signature.
mdp	string	<p>(Acrobat 7.0) The modification, detection, and prevention (MDP) setting to use when signing the field. Values include:</p> <ul style="list-style-type: none"> allowNone default defaultAndComments <p>While allowAll is a legal value, it cancels out the effect of mdp and no certification signature can be used for this field.</p>

Table 8 Seed values: Object properties and descriptions

Property	Type	Description
reasons	array of strings	A list of reasons that the user is allowed to use when signing. (Acrobat 8.0) If this array contains a single empty string and reasons are marked as required using the flags variable, Acrobat will not allow a signing reason. If this array is empty and reasons are marked as required, an exception will be thrown.
shouldAddRev Info	boolean	(Acrobat 8.0) The default value is false. If true, the application does certificate and chain revocation checking and embeds the information in the signature. If true and the flag is set to require these actions, any failure in these actions results in signing failure. Only relevant if subFilter is adbe.pkcs7.detached or adbe.pkcs7.sha1. If the subFilter is adbe.x509.rsa_sha1 and adding revocation information is required, the signing operation fails.
subFilter	array of strings	An array of acceptable formats to use for the signature. Refer to the Signature Info object's subFilter property for a list of known formats.
timeStampSpec	object	(Acrobat 7.0) A seed value timeStamp specifier object. It uses the url and flags properties to specify a timestamp server. For details, see "Specifying Timestamps for Signing" on page 76
version	number	The minimum required version number of the signature handler to be used to sign the signature field. Valid values are 1 and 2. (Acrobat 8) This must be set to 2 if this seed value object contains any Acrobat 8-specific content that is marked as required.
lockDocument	name	(Optional; PDF 1.7-ADBE-3) Allows the author to add a Lock Document checkbox to the signing dialog so a signer can lock the document at the time of signing. The default is auto. <ul style="list-style-type: none"> • true: Indicates that the desired action is that the document should be locked at the time of signing. If the Ff entry indicates that LockDocument is not a required constraint and that the user can choose to override this at the time of signing. Otherwise the document must be locked after signing. • false: A false value indicates that the document should not be locked after signing. Again, the required flag determines whether this is a required constraint. • auto: The auto value allows the consuming application to decide whether or not to present the lock UI for the document and whether to honor the required flag based on the properties of the document.
appearanceFilter	string	(Optional; PDF 1.7-ADBE-3) A text string naming the appearance to be used when signing the signature field. Conforming readers may choose to maintain a list of named signature appearances and this text string provides authors with a means of specifying which appearance should be used to sign the signature field. If the required bit in Ff is set (see flag above), then the appearance must be available to sign the document and must be used.

5.1.3 Enabling JavaScript to Set Seed Values

Authors sometimes use JavaScript to set seed values for signature fields. When Acrobat's JavaScript console is used for JavaScript execution, the JavaScript debugger must be enabled.

Tip: If you do not intend to set seed values with JavaScript through Acrobat's JavaScript debugger, skip this section.

To enable the JavaScript debugger:

1. Choose **Edit > Preferences** (Windows) or **Acrobat > Preferences** (Macintosh).
2. Choose **JavaScript** in the left-hand category list.

3. Check **Enable JavaScript**.
4. Check **Enable JavaScript debugger after Acrobat is restarted**.
5. Restart Acrobat.

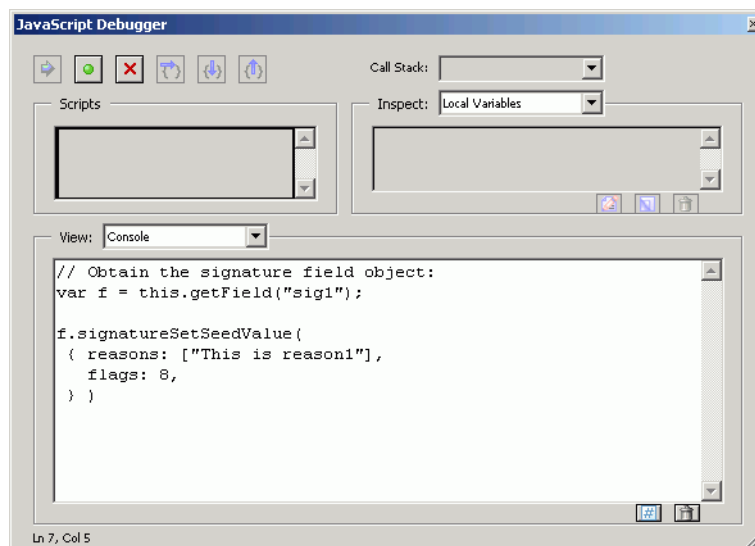
To set seed values with the console (JavaScript debugger) in Acrobat, do the following:

1. Choose **Ctrl + J**.
2. Use the **View** drop-down list and select Console.
3. Enter the requisite JavaScript.
4. Highlight the JavaScript. If you do not highlight the JavaScript, only the last line of code is executed.
5. Press **Control + Enter** simultaneously or select the **Enter** key on the numeric keypad.

Tip: When the JavaScript is executed correctly, the debugger returns “undefined.”

6. Save the document, and test the field.

Figure 55 Seed values: JavaScript debugger



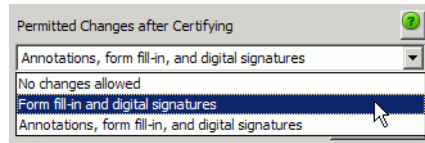
5.2 Forcing a Certification Signature

By default, signature fields can be signed with an approval or certification signature at the time of signing. However it is possible to constrain a signature field such that only a certification signature can be used.

Certification signatures are always associated with modification detection and prevention settings (and an `mdp` property) that control what types of changes can be made to a document before the signature becomes invalid. Changes are stored in the document as incremental saves beyond the original version of the document that was covered by the certifying signature. The `mdp` seed value allows you to control what behavior the signer can allow after signing (Figure 56).

Note: If a document is already signed, fields with the `mdp` property specified will NOT invoke the certifying workflow. No error is given. Do not use `mdp` unless you are sure the requisite field will be the first one signed.

Figure 56 Seed value: Forcing mdp selection during certification



MDP has one of the following four values:

- **allowAll:** Do not use `allowAll` unless you want to force an *approval* signature since this value results in MDP not being used for the signature and therefore doesn't force a certifying signature.
- **allowNone:** Document changes invalidate the signature and lock the author's signature. `allowNone` bypasses any custom `legalAttestations` because no document changes can occur and the user does not therefore need to be warned about malicious content. Do not use with `legalAttestations`.
- **default:** Allow form field fill-in if fields are present in the document as well as additional signatures. Other changes to the document invalidates the signature.
- **defaultAndComments:** Allow form field fill-in if fields are present in the document and allows annotations (comments) to be added, deleted, or modified as well as additional signatures. Other changes to the document invalidates the signature. Note that annotations can be used to obscure portions of a document and thereby affect the visual presentation of the document.

To force a certifying signature for a particular field:

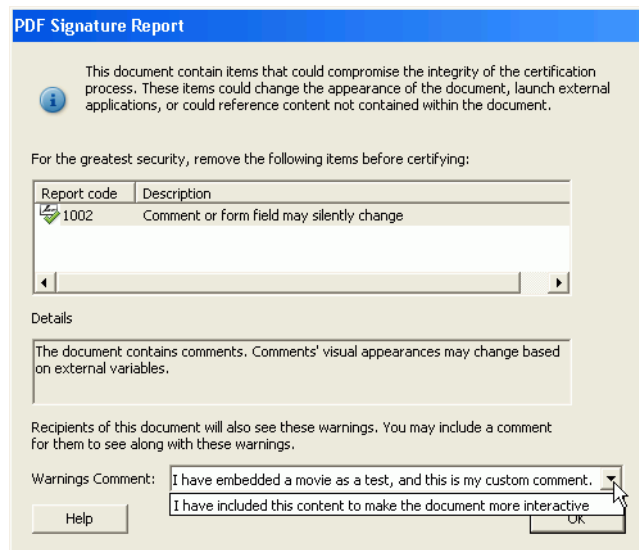
1. Create a signature field with an intuitive name and tooltip.
2. Create the JavaScript that gets the field object and uses the seed value method ([Example 5.1](#)).
3. Set the `mdp` value:
 - **allowNone:** Do not use with `legalAttestations`.
 - **default:** Allow form field fill-in, including signing.
 - **defaultAndComments:** Allow form field fill-in, signing, and comments.
4. Add `legalAttestations` if you would like to customize user choices.

Signers can view warnings about potentially malicious content (content that could change the appearance of a signed document) during signing. The Review button in the signing dialog runs the PDF/SigQ Conformance Checker which reports on rich content. Signers can then enter a **Warnings Comment** in the drop-down list indicating why that content is OK.

When specifying custom legal attestations, keep the following in mind:

- Since certified document warnings only appear in certifying workflows, only use `legalAttestations` if you also use `mdp`. For details, see ["Forcing a Certification Signature" on page 71](#).
- Customizing legal attestations overrides and removes default choices for the signer.
- Custom text is viewable in the user interface during signing when the signer chooses **Review** in the signing dialog.

Figure 57 Seed values: Custom legal attestations



5. Highlight the JavaScript and choose **Control + Enter** or choose the **Enter** key on the numeric keypad. When someone signs the field, the certifying workflow is invoked and only the specified `mdp` settings will be available (Figure 56).
6. Run the JavaScript, save the document, and test the field.

Example 5.1 Seed value: `mdp`

```
// Obtain the signature field object:
var f = this.getField("mySigFieldName");

f.signatureSetSeedValue(
{
    mdp: "defaultAndComments",
    legalAttestations: ["Approved by Management", "Signed by Procurement"]
} )
```

5.3 Giving Signers the Option to Lock a Document

While certifying a document allows authors and signers to restrict certain document features, for example, subsequent signing and filling out forms, these permissions are set at the document level and cannot become more restrictive as signatures are applied. Acrobat 9 provides a seed value that adds a **Lock Document** checkbox to the signing dialog when the configured field is selected for signing. This allows any signer to completely lock the document from further changes.

The author is given the choice of requiring their seed value to be honored by the signing application or to be an optional recommendation. This is provided through the use of the required flags bitfield to indicate whether or not the seed value is optional or mandatory. The options are as follows:

- **true:** A value of true indicates that the desired action is that the document should be locked at the time of signing. If the `Ff` entry indicates that `LockDocument` is not a required constraint, the user can choose to override this at the time of signing. Otherwise the document must be locked after signing.

- **false:** A false value indicates that the document should not be locked after signing. By default, the user is given the option of locking after signing. Again, the required flag determines whether this is a required constraint.
- **auto:** Default. The auto value allows the consuming application to decide whether or not to present the lock UI for the document and whether to honor the required flag based on the properties of the document.

Example 5.2 Seed value: lockDocument

```
f = this.getField("mySigFieldName");  
f.signatureSetSeedValue({lockDocument:<true/false/auto>});  
  
//Set the setting as required  
f.signatureSetSeedValue({lockDocument:<true/false/auto>, flags:0x80});
```

Figure 58 Sign Document dialog: With Lock Document checkbox added

Sign Document

Sign As: Mr. Example <example@moosepathbandb.com>

Password: *****

Certificate Issuer: Mr. Example Info...

Appearance: Standard Text

Mr. Example

Digitally signed by Mr. Example
DN: cn=Mr. Example,
o=moosepathbandb,
ou=moosepathbandb,
email=example@moosepathbandb.com, c=US
Reason: This is a reason
Date: 2008.08.22 13:29:33 -0700

☒ Lock Document After Signing

Additional Signature Information

Reason: This is a reason

Location:

Contact Info:

Click Review to see if document content may affect signing Review...

Sign Cancel

5.4 Forcing Signers to Use a Specific Signature Appearance

Enterprises and other structured work environments sometimes provide users with predefined signature appearances. These appearances are then used for specific signing purposes. For example, the appearance may identify the signer's organizational affiliation or a particular task or workflow that pertains to the signed document.

Authors in such environments can specify which signature appearance is required for any given signature field. As with other seed values, a flag bit is used to indicate whether or not the field is a recommendation or mandatory. A signature field is correlated with a specified name which is then used to match a given appearance. The string name must exactly match the name of a signature appearance for it to be selected.

Example 5.3 Seed value: signatureAppearance

```
f = this.getField(<Field Name>);
f.signatureSetSeedValue({appearanceFilter:"Example Appearance Name"});

//Set the setting as required
f.signatureSetSeedValue({appearanceFilter:"Example Appearance Name", flags:0x100});
```

5.5 Adding Custom Signing Reasons

Acrobat predefines several common signing reasons such as "I am approving this document." However, the author can specify custom reasons and make those reasons required or optional. When custom reasons are marked as required, users cannot enter any new reasons as the field becomes read-only. When those reasons are flagged as optional, signers can choose one of the provided reasons or create a new one by typing in the **Reason** field. Specifying a signing reason will remove all of the default reasons from the reason drop-down list.

User interface impact: Note that end users have a user interface preferences that allows them control whether or not the reason's field appears. The preference interacts with the reasons flag as shown in [Table 9](#), and the logic is as follows:

- The document author has control over whether the UI appears and the `required` flag overrides user-specified settings.
- When a flag makes the field `optional`, end users can enter custom reasons.

To specify custom signing reasons:

1. Create a signature field with an intuitive name and tooltip.
2. Create the JavaScript that gets the field object and uses the seed value method ([Example 5.4](#)).
3. Add the reasons. The reason list is an array in the format of ["one", "two", "three"].
4. Enter a flag value to indicate whether the value is required or not.
 - If a reason is not required, signers can add their own custom reason while signing.
 - If the predefined reasons are required, signers are prevented from saving a document with their own reason ([Figure 59](#)).
5. Run the JavaScript, save the document, and test the field.

Table 9 Reason field behavior

# of Reasons	UI Pref	Flag	Reason Behavior
0 (empty array)	off	Required	Reason field does not appear in UI.

Table 9 Reason field behavior

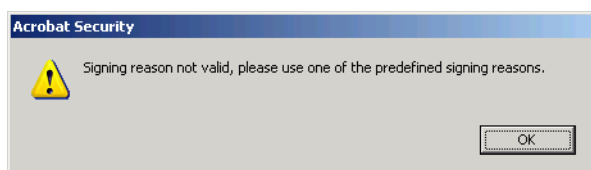
# of Reasons	UI Pref	Flag	Reason Behavior
0 (empty array)	on	Required	Reason field does not appear in UI.
0 (empty array)	off	Optional	Reason field does not appear in UI.
0 (empty array)	on	Optional	Display the default list.
1 or more	off or on	Required	Display the custom reasons in a read-only field.
1	off	Optional	Reason field does not appear in UI.
2 or more	on	Optional	Display the custom drop-down list and let the user enter a custom reason.
2 or more	off	Optional	Reason field does not appear in UI.

Example 5.4 Seed value: Custom signing reason

```
// Obtain the signature field object:
var f = this.getField("mySigFieldName");

f.signatureSetSeedValue(
{ reasons: ["This is a reason", "This is a better reason"],
  flags: 8
} )
```

Figure 59 Seed value: Reason not allowed error



5.6 Specifying Timestamps for Signing

Timestamps originating from a timestamp authority's timestamp server are often associated with signatures. If it is critical in your workflow to acquire a secure timestamp with a digital signature, it can be controlled at the document level instead of relying on the signer's Acrobat configuration. Adding a seed value to the signature field with the timestamp server authority settings overrides the corresponding application level settings, if any. Use the `timeStampsSpec` specifier object's `url` and `flags` properties to specify a timestamp server.

Table 10 Seed values: timeStampsSpec properties

Property	Type	Description
<code>url</code>	string	URL of the timeStamps server providing a RFC 3161-compliant timeStamps.
<code>flags</code>	number	A flag controlling whether the time stamp is required (1) or not required (0). The default is 0.

To specify a timestamp server:

1. Create a signature field with an intuitive name and tooltip.

2. Create the JavaScript that gets the field object and uses the seed value method ([Example 5.5](#)).
3. Provide a URL for the `timeStampsSpec` object.

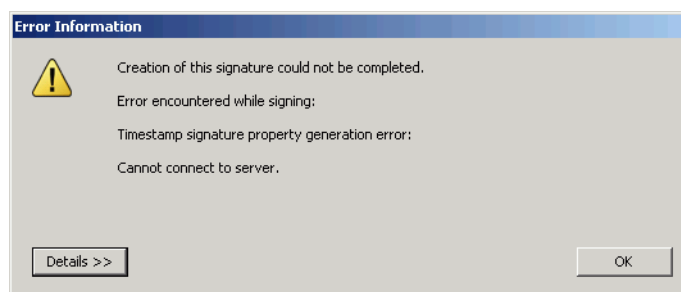
Tip: Timestamp seed value settings override the end users' application settings, if any.
4. Enter a flag value to indicate whether the value is required or not.
 - If it is required, the field is automatically timestamped on signing. If the application cannot find the server, an error appears ([Figure 60](#)).
 - If it is not required, the field will is automatically timestamped on signing if the application can find the server. If it cannot find the server, the signature is not timestamped and no error appears.
5. Run the JavaScript, save the document, and test the field.

Example 5.5 Timestamp server seed value

```
// Obtain the signature field object:
var f = this.getField("mySigFieldName");

f.signatureSetSeedValue(
{
  timeStampsSpec: {
    url: "http://153.32.69.130/tsa",
    flags: 1
  }
} )
```

Figure 60 Time stamp server error



5.7 Specifying Alternate Signature Handlers and Formats

Organizations may choose to use alternate signature technologies or implementations (signature handlers), provided by third party software developers. For example, a corporation may have deployed Entrust Entelligence® to all their desktops and may choose to use the Entrust signature plug-in with Acrobat. Two seed values allow authors to specify which signature handler and format to use. By using a standard format, interoperability across multiple signature handlers is possible.

Filter also allows authors to control what handler version is required. For example, for Acrobat 6.x, the PPKLite version is 0. For Acrobat 7.x, the PPKLite version is 1. Therefore, specifying a version of 1 prevents signers from signing when their application is older than Acrobat 7.0. Custom handlers can use any version as required.

User interface impact: Specifying a non-default handler can result in a different user interface and workflow during signing.

Seed values for specifying handlers and signature types are the following:

- **filter:** `filter` is the internal name of a signature handler. Signature handlers perform a number of functions including signature validation. While Acrobat ships with a default handler (Adobe.PPKLite), custom or third-party handlers such as those from Entrust and VeriSign may be used. The Acrobat SDK describes how to write a custom handler (Adobe.DocSign).

Tip: `filter` is often used in conjunction with `version` when a minimum filter version is required.

- **subfilter:** `subfilter` is the internal name of the signature format, such as `adbe.pkcs7.detached` intended to be verifiable by signature handlers other than the one that created it. Signature handlers need to be able to understand the signature type (or format).

Tip: Since it is possible that different handlers might be used for signing and validating, `filter` and `subfilter` are used together to assure that signing workflows with different components are interoperable. These properties are identical to those in the signature dictionary. For more information, refer to the **PDF Reference**.

To specify a signature handlers and format type:

1. Create a signature field with an intuitive name and tooltip.
2. Create the JavaScript that gets the field object and uses the seed value method ([Example 5.6](#)).
3. Specify a `filter`.
4. If `filter` is specified, you may use the optional `version` as follows:
 - PPKLite for Acrobat 6.X: 0
 - PPKLite for Acrobat 7.x: 1
 - Custom handlers: Any.
5. Enter the handler name and `subfilter` type. Third parties may define their own subfilters but should follow the naming convention recommended in the **PDF Reference**. The **PDF Reference** defines the following standard `subfilter` values:
 - `adbe.x509.rsa_sha1`
 - `adbe.pkcs7.detached`
 - `adbe.pkcs7.sha1`
6. Run the JavaScript, save the document, and test the field.

Example 5.6 Seed value: Specifying signature components

```
// Obtain the signature field object:
var f = this.getField("mySigFieldName");

f.signatureSetSeedValue(
{
  filter: "Entrust.PPKEF",
  subfilter: "adbe.x509.rsa_sha1"
} )
```

5.8 Specifying a Signature Hash Algorithm

When a signer's digital ID contains RSA public and private keys, it is possible to specify alternative signature hash algorithms. The default algorithm is SHA1, and the alternatives are listed in [Table 8](#).

User interface impact: Once a document is signed, the signature's hash algorithm can be viewed by right clicking on a signature, choosing **Show Signature Properties**, and displaying the Document tab. The algorithm is displayed in the Hash Algorithm field.

Caution: If a signer may be using FIPS mode, do NOT specify MD5 or RIPMD160.

To specify a non-default algorithm:

1. Create a signature field with an intuitive name and tooltip.
2. Create the JavaScript that gets the field object and uses the seed value method ([Example 5.7](#)).
3. Specify the `digestMethod`. This can be an array of comma-separated items such as `['RIPMD160', 'SHA384']`.
4. Run the JavaScript, save the document, and test the field.

Example 5.7 Hash algorithm seed value

```
// Obtain the signature field object:
var f = this.getField("mySigFieldName");

f.signatureSetSeedValue({
    digestMethod: ['SHA384']
});
```

5.9 Embedding Revocation Information in a Signature

Users (signers) have the option to embed certificate revocation status in a signature by turning on **Include signature's revocation status when signing** in their preferences. However, the default value is false (revocation information is not embedded), and document authors may need to force embedding of revocation information regardless of the users application settings. Embedding the signing certificate's revocation status in a document allows recipients to validate certificates (signatures) while offline and speeds up the revocation checking process. Moreover, if a certificate is revoked or expired at some time after signing, embedded revocation information enables the application to determine if a certificate was valid at the time of signing so that the signature status will remain valid.

Note: Only relevant if `subFilter` is `adbe.pkcs7.detached` or `adbe.pkcs7.sha1`. If the `subFilter` is `adbe.x509.rsa_sha1` and adding revocation information is required, the signing operation fails.

To force embedding of certificate revocation information in a signature:

1. Create a signature field with an intuitive name and tooltip.
2. Create the JavaScript that gets the field object and uses the seed value method ([Example 5.7](#)).
3. Set `shouldAddRevInfo` to true.

4. Run the JavaScript, save the document, and test the field.

Example 5.8 Hash algorithm seed value

```
// Obtain the signature field object:  
var f = this.getField("mySigFieldName");  
  
f.signatureSetSeedValue({  
    shouldAddRevInfo: true  
});
```

5.10 Specifying Certificate Properties for Signing

Certificate seed values are commonly used to restrict signing to particular certificates such as those issued by particular certificate authorities or containing numbers that specify certain policies with “object identifiers” or “OIDs.” Authors specify which certificate signers must use by setting the certSpec object’s properties (Table 11). These can be preferences or requirements. If a certificate cannot be found that matches a required certificate seed value, a URL can be provided to allow the signer to get more information such as how to obtain an appropriate certificate.

Certificate specification can be used to streamline workflows. When one certificate is allowed, the digital ID dialog is bypassed and the signer is directed to sign and save immediately. Signing fails if the selected certificate is not an exact match. It is also often expedient to provide a URL value so that users are directed to a help page or some location where a digital ID can be obtained.

Figure 61 Seed value: Specifying certificates for signing

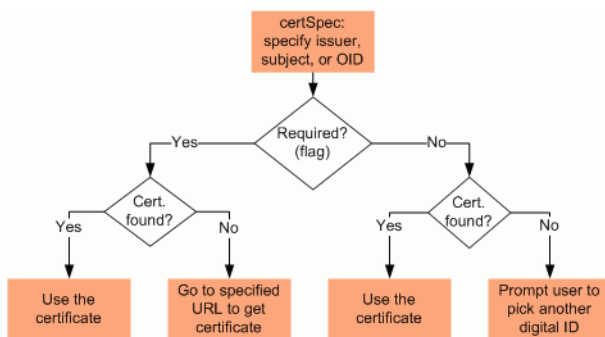


Table 11 Seed values: certSpec properties

Property	Type	Description
flags	number	<p>A set of bit flags controlling which of the following properties of this object are required. The value is the logical OR of the following values, which are set if the corresponding property is required:</p> <p>1: subject</p> <p>2: issuer</p> <p>4: oid</p> <p>8: subjectDN (Acrobat 8 and later)</p> <p>16: issuerDN (Acrobat 8 and later)</p> <p>32: keyUsage (Acrobat 8 and later)</p> <p>64: url (Acrobat 8 and later)</p> <p>If this field is not present, all properties are optional.</p> <p>Usage: 1 specifies <code>subject</code>, 3 specifies <code>subject</code> and <code>issuer</code>, and 6 specifies <code>issuer</code> and <code>oid</code>. That is, values can be added. If this field is not present, all properties are optional.</p>
issuer	array of certificate objects	<p>One or more issuers that are acceptable for signing. The issuer can be a root or intermediate root certificate. Access to the physical, DER-encoded certificate is required. It is identified by a path to a discrete file in the format of <code>["/c/test/root.cer"]</code>.</p>
keyUsage	array of integers	<p>(Acrobat 8.0) Integers in HEX or decimal that specify the <code>keyUsage</code> extension that must be present in the signing certificate. Each integer is constructed as follows:</p> <p>There are two bits used for each <code>keyUsage</code> type (defined in RFC 3280) starting from the least significant bit:</p> <p><code>digitalSignature</code>(bits 2,1)</p> <p><code>nonRepudiation</code>(4,3)</p> <p><code>keyEncipherment</code>(6,5)</p> <p><code>dataEncipherment</code>(8,7)</p> <p><code>keyAgreement</code>(10,9)</p> <p><code>keyCertSign</code>(12,11)</p> <p><code>cRLSign</code>(14,13)</p> <p><code>encipherOnly</code>(16,15)</p> <p><code>decipherOnly</code>(18,17)</p> <p>The value of the two bits have the following semantics:</p> <p>00: The corresponding <code>keyUsage</code> is not allowed.</p> <p>01: The corresponding <code>keyUsage</code> is required.</p> <p>10 and 11: The state of the corresponding <code>keyUsage</code> doesn't matter.</p> <p>For example, if it's required that <code>keyUsage</code> must require <code>digitalSignature</code> and the state of all other's doesn't matter, then the corresponding integer would be <code>0x7FFFFFFD</code>. That is, to represent <code>digitalSignature</code>, set 01 for bits 2 and 1 respectively, and set 11 for all other <code>keyUsage</code> types.</p>
oid	array of strings	<p>One or more policy OIDs that must be present in the signing certificate's policy. The OID is part of the value of the certificate's certificate policy field. This property is only applicable if the <code>issuer</code> property is present. <code>oid</code> and <code>issuer</code> can be used together to specify a certificate that has the selected policy.</p>

Table 11 Seed values: certSpec properties

Property	Type	Description
subject	array of certificate objects	One or more subjects that are acceptable for signing. The subject property identifies specific individuals (as certificate owners) that can sign. Access to the physical, DER-encoded certificate is required. It is identified by a path to a discrete file in the format of <code>["/c/test/root.cer"]</code> .
subjectDN	array of certificate objects	(Acrobat 8.0) Each object specifies a subject distinguished name (DN) acceptable for signing. More than one DN may be specified, but a signing certificate must satisfy at least one of the DNs by containing all the attributes specified in the matching DN. DN attribute restrictions are specified by adding them as properties. The properties' key names can either be the corresponding attributes' friendly names or OIDs (as defined in RFC 3280). The properties' value must be of type string. For more information about the various attributes and their types, refer to RFC 3280.
url	string	A URL that can be used to enroll for a new certificate if a matching one is not found, such as <code>https://aardvark.corp.example.com/</code> . Works in conjunction with <code>urlType</code> (if present). A degenerate use of this property is when the URL points to a Web service that is a digital ID store such as a roaming ID server. In that case, the URL indicates that as long as the signer has a digital ID from that Web service, it is acceptable for signing.
urlType	string	(Acrobat 8.0) The <code>url</code> type. If this attribute isn't present, it's assumed that the <code>url</code> points to a HTML site. There are two supported types: HTML: An HTML website. Acrobat uses the Web browser to display its contents. ASSP: A URL to a web service using the ASSP protocol for roaming ID servers.

5.10.1 Specifying Signing Certificates Origin

Authors can limit potential signers to individuals or groups as follows:

- `subject` limits potential signers to only those specified individuals. Signers could be limited to one or more people.
- `issuer` limits signers to those with certificates that chain up to a common, shared issuer. For example, all of a company's employees may use the company's certificate as an intermediate certificate and that certificate could be used as the issuer.
- `subjectDN` limits signers to those with certificates that match all the attributes of one of the listed DNs. For example:
 - `{cn:"Alice", ou:"Engineering", o:"Acme Inc"}`. For details about the friendly names of DN attributes (cn, o, ou, and so on), refer to the RDN Object in the *JavaScript for Acrobat API Reference*.
 - `{cn:"Joe Smith", ou:"Engineering", 2.5.4.43:"JS"}`, where OID 2.5.4.43 is used to carry out matching for the "Initials" attribute.

The following is sample code to define the above DN:

```
var subjectDN = {cn:"Joe Smith", ou:"Engineering"};
subjectDN["2.5.4.43"] = "JS";
```

Attributes whose value is of type `DirectoryString` or `IA5String` can be specified as shown in the example above, whereas all other value types, e.g. `dateOfBirth` whose value is of type `GeneralizedTime`, the value needs to be specified as a hex encoded binary string.

To specify a certificate:

1. Create a signature field with an intuitive name and tooltip.
2. Get the required certificates and install them in some accessible location.
Tip: They must be in a .cer files in a DER format.
3. Create the JavaScript that gets the field object and uses the seed value method. Use `security.importFromFile` to get the DER- encoded certificates from their installed location (Example 5.9).
4. Add the `subject` and `issuer` properties to the `certspec` object.
5. Enter a flag value to indicate whether the value is required or not. Either or both the `subject` and `issuer` may be required.
6. Run the JavaScript, save the document, and test the field.

Example 5.9 Certificate issuer and subject seed value

```
// Obtain the signature field object:
var f = this.getField("mySigFieldName");

var mySubjectCert = security.importFromFile("Certificate",
"/C/Temp/nebwhifflesnit_DER.cer");
var myIssuerCert = security.importFromFile("Certificate",
"/C/Temp/nebsCompany_DER.cer");

f.signatureSetSeedValue(
{
  certspec: {
    subject: [mySubjectCert],
    issuer: [myIssuerCert],
    flags: 3
  }
} )
```

5.10.2 Specifying Certificates by Key Usage

Acrobat's default signature handler allows signing with certificates where the **Key usage** field is *Sign transaction* or *Sign document*. However, the `keyUsage` seed value allows you to override the default behavior and limit signing to those certificates where the `keyUsage` is set to any value defined in RFC 3280 (see Table 11). While the seed value could be used to require or disallow any of RFC 3280 `keyUsage` values, the two most common cases allow or disallow `digitalSignature(bits 2,1)` (displayed as *Sign transaction* in Acrobat's Certificate Viewer) or `nonRepudiation(4,3)` (displayed as *Sign document* in Acrobat's Certificate Viewer). However, any combination of uses may be set.

To restrict signing to a certificate with a particular `keyUsage`:

1. Create a signature field with an intuitive name and tooltip.
2. Create the JavaScript that gets the field object and uses the seed value method (Example 5.10).
3. Specify the `keyUsage` value in HEX:

1. Specify 00, 01, 10, or 11 for each of the `keyUsage` values beginning with the least significant bit (the last one in the list in [Table 11](#)). For example:
 - `digitalSignature` is disallowed and non repudiation is required, and other values don't matter: 11111111111110100. Convert to HEX: 3FFF4
 - `digitalSignature` is required and non repudiation is disallowed, and other values don't matter: 11111111111110001. Convert to HEX: 3FFF1
2. Remove the 3 and prepend the HEX value with 0x7FFF so it is in the correct HEX 32-bit format such as 0x7FFFFFFF1.
3. Enter a flag value to indicate whether the value is required or not. Set 32 if `keyUsage` is required and there are no other `certspec` properties.
4. Run the JavaScript, save the document, and test the field.

Example 5.10 Certificate key usage seed value

```
// Obtain the signature field object:
var f = this.getField("mySigFieldName");

f.signatureSetSeedValue({
  certspec: {
    keyUsage: [0x7FFFFFFF1], //Set KeyUsage to "digitalSignature"
    flags: 32 //Require keyUsage
  },
});
```

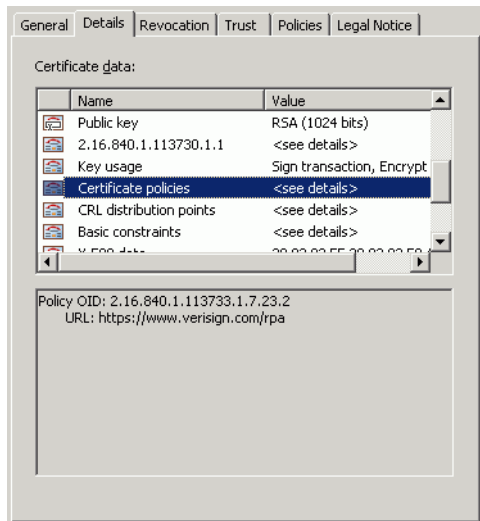
5.10.3 Specifying Certificates by Policy

For legal reasons, policies are often associated with certificates. One way policies are identified is through an object identifier (OID), a unique series of numbers in the certificate policies' field that identifies the policy. Since an `oid` is always used with the `issuer`, authors can use this seed value pair when a company issues different certificates with different policies and it is necessary to restrict signing to certificates associated with a certain policy.

To restrict signing to a certificate containing a specific policy:

1. Create a signature field with an intuitive name and tooltip.
2. Create the JavaScript that gets the field object and uses the seed value method ([Example 5.11](#)).
3. Specify the `issuer`.
4. Specify the `oid`. A policy OID is part of the value of the certificate's certificate policy field ([Figure 62](#)).
5. Enter a flag value to indicate whether the value is required or not. A value of 6 is recommended since `issuer` and `oid` must be specified together.
6. Run the JavaScript, save the document, and test the field.

Figure 62 Policy OID



Example 5.11 Certificate policy seed value

```
var myIssuerCert = security.importFromFile("Certificate",  
"/C/Temp/nebsCompany_DER.cer");  
  
// Obtain the signature field object:  
var f = this.getField("mySigFieldName");  
  
f.signatureSetSeedValue(  
  {  
    certspec: {  
      issuer: [myIssuerCert],  
      oid: ["2.16.840.1.1.113733.1.7.23.2"],  
      flags: 6  
    }  
  }  
);
```

5.10.4 Specifying a URL When a Valid Certificate is not Found

When a valid certificate is not found, users can be redirected to a URL during the signing workflow. The URL may be to a server with a certificate repository; or, more likely, the URL may be a link to a Web page describing how to obtain a new or valid certificate.

To specify a certificate with a specific policy:

1. Create a signature field with an intuitive name and tooltip.
2. Create the JavaScript that gets the field object and uses the seed value method ([Example 5.12](#)).
3. Specify a certificate as described in one of the previous sections. Use `issuer` and/or `subject`.
4. Specify the URL. The URL can point to a certificate server or to instructions for getting a certificate.
5. Run the JavaScript, save the document, and test the field.

Example 5.12 Alternate certificate URL seed value

```
// Obtain the signature field object:
var f = this.getField("mySigFieldName");

var mySubjectCert = security.importFromFile("Certificate",
"/C/Temp/nebwhifflesnit_DER.cer");

f.signatureSetSeedValue(
{
  certspec: {
    subject: [mySubjectCert],
    url: "https://aardvark.corp.example.com/",
  }
} )
```

5.10.5 Restricting Signing to a Roaming ID

Fields can be required to be signed with roaming IDs by specifying the `certspec url` and `urlType` properties. By providing the roaming ID server URL and the ASSP protocol as arguments, only roaming IDs associated with the specified server will appear in the signing dialog's digital ID drop-down list when a user attempts to sign the field.

To require signing only with a roaming ID:

1. Create a signature field with an intuitive name and tooltip.
2. Create the JavaScript that gets the field object and uses the seed value method ([Example 5.13](#)).
3. Specify the roaming ID server URL.
4. Specify ASSP as the URL type.
5. Run the JavaScript, save the document, and test the field.

Example 5.13 Roaming ID seed value

```
// Obtain the signature field object:
var f = this.getField("mySigFieldName");

f.signatureSetSeedValue(
{
  certspec: {
    url: "https://myroamingIDserver.arcot.com:9777",
    urlType: "ASSP",
  }
} )
```

5.11 Custom Workflows and Beyond

Advanced document and workflow customization is beyond the scope of this document. However, keep in mind that Acrobat's security APIs allow users many opportunities for customization. Document developers

can easily create custom signing menu items, automate tasks, and perform other operations beyond those described in the preceding seed value sections.

For example, [Example 5.14](#) performs a number of operations that would simplify signing operations in an enterprise setting. The script adds a **Request Employee Signature** to the toolbar and set up a number of automatic actions. When a user selects the menu item, a signature field with predefined properties is automatically created in the needed document location, and the field's seed values are set.

Note: For more information, refer to the online *Acrobat JavaScript Scripting Guide*, *JavaScript for Acrobat API Reference*, *PDF Reference*, and the Acrobat SDK.

Example 5.14 Automating signing tasks

```
//*****
//File: seedValue.js
//Purpose: Demo how to set certificate constrictions into a signature field
//Steps: 1. Add a menu item under Tools, called Request Employee Signature
//        2. Add a signature and text field(for display)to the current open file
//        3. Set seed value
//           3.1 Wrap certificate object
//           3.2 Set seed value to the added signature field
//              reason: "I am approving this document"
//              certSpec:
//                  issuer: Example/MyCompanyCA (the root)
//                  oid: 2.5.29.16 the oid of Example/MyCompanyCA
//                  url: https://my.corp.example.com/
//                  flag: 2 set limits on issuer
//        4. Display seed value added to the sig field to the added text field.
//*****
// 1. Add a Tools menu item called Request Employee Signature
app.addItem
({
  cName: "Request Employee Signature",
  cParent: "Tools",
  cExec: "setSeedValues()",
  cEnable: "event.rc = (event.target != null);",
  nPos: 0
});

//Run function when menu item "Request Employee Signature" is clicked
function setSeedValues(){
//modify the following according needs
  var sigfieldName = "aSigField";
  var myReasons = ["I am approving this document"];
  var myIssuer;
  var oids = ["2.5.29.16"];
  var url = "https://seneca.corp.example.com/";
  var certSpecFlag = 2;//constricts on issuer, 6: issuer + OID, 1: users to
sign, 7: issuer + oid + user
  var svFlag = 0;      //no restrictions

  try{
    //2. add a sig field called "aSigField" and a text field
    var field =
this.addField(sigfieldName,"signature",0,[180,640,352,680]);//1st page
```

```
        field.borderStyle = border.s;
        field.fillColor = color.ltGray;
//a text field to display what seed values set to the sig field
        var textField = this.addField("aText", "text", 0, [110,360,500,550]);
        textField.borderStyle = border.s;
        textField.fillColor = color.yellow;
        textField.multiline = true;
        textField.display = display.hidden; //hidden form screen and print
        textField.setAction("MouseUp", "event.target.display =
display.hidden;");//click field, field disappears

//3. set seed value
//3.1 set up issuer's certificate object
        var myissuerDN = {CN:"Enterprise Services CA", OU:"VeriSign Trust
Network", O:"Example Systems Incorporated"};
        var mykeyUsage = ["kDigitalSignature","kCRLSign"];
        var myMD5Hash = "BF70 913F F8D6 D60A 47FE 8253, 3081 5DB4";
        var mySHA1Hash = "6b e8 46 06 39 f5 65 18 48 b2 f8 3a b1 46 3f 56 02 be 06
c3";
        var myserialNumber = "3e 1c bd 28";
        var mysubjectCN = "Example Root CA";
        var mysubjectDN = {CN:"Example Root CA",OU: "Example Trust Services",O:
"Example Systems Incorporated",C:"US"};
        var myusage = {endUserSigning:true};
        var ExampleRootCertBinary =
"308204A130820389A00302010202043E1CBD28300D06092A864886F70D01010505003069310
//-----<snip>-----
440512D9E9B47DB42A57C1FC2A648B0D7BE92694DA4F62957C5781118DC8751CA13B2629D4F2
B32BD31A5C1FA52AB0588C8";

//var myIssuer = {binary:ExampleRootCertBinary,issuerDN:myissuerDN,keyUsage:
mykeyUsage,MD5Hash:myMD5Hash,
//SHA1Hash:mySHA1Hash,serialNumber:myserialNumber,subjectCN:
mysubjectCN,subjectDN:mysubjectDN,usage:myusage};
        var myIssuer = security.importFromFile("Certificate",
"/c/test/root.cer");//if import from an external reference
// 3.2 set up seed value
        field.signatureSetSeedValue({reasons:myReasons,certspec:{ issuer:
[myIssuer],/*oid:oids,*/url:url,flags:certSpecFlag}, flags:svFlag});
//4. Display seed value added to the signature field to the new text field
        var result = "";
        var w = field.signatureGetSeedValue();
        for(i in w)
            result += ( i + " = " + eval("w." +i) + "\n");

        var z = w.certspec;
        for(i in z)
            result += ( i + " = " + eval("z." +i) + "\n");

        textField.value = result + "** Click on me to make me disappear **";
        textField.display = display.show; //display what seed values were set
    }catch (e){
        app.alert("setSeedValues(): " + e );
    }
}
```

Like a conventional, handwritten signature, digital signatures identify the signer. However, digital signatures also enhance security because they store information about the signer as well as the signed document. For example, signatures can be used to verify signed content has not been altered, confirm the signer's identity and to prevent the signer from denying their own signature. Before signing, review the [Signing Basics](#) and then refer to the following:

- ["Signing With a Certification Signature" on page 90](#)
- ["Signing with an Approval Signature" on page 96](#)

6.1 Signing Basics

6.1.1 Before You Sign ...

Before signing, do the following:

- **Configure the signing application:** Both authors and signers should configure their application environment. For details, see ["Setting up the Signing Environment" on page 46](#).
- **Obtain a digital ID:** Get a digital ID from your own organization, a 3rd-party provider or create a self-signed one.
- **Finish editing the document:** Sign only after making final changes. Post-signing changes may impact signature validity.
- **Pick a signature type:** Learn about approval and certification signatures so you know which to use.

6.1.2 Signature Types

A document can contain certification and/or approval signatures. Which signature type you need depends on the intent of both the author and the signer. Signature types include the following:

- **Certification Signature:** A certification signature provides a higher level of document control than an approval signature. Because it must be the first signature in a document, certification menu options are disabled if another signature is already present. Certified documents that have not been invalidated by illegal changes may display a blue ribbon icon next to the digital signature ([Figure 65](#)). Use certification signatures for the following:
 - When you as the document author want to attest to the document contents.
 - When you want to restrict the actions of future document recipients.
 - For documents that will be signed multiple times. You can specifically permit additional signatures so that the status of existing signatures is not impaired as signatures are added.
 - When you as the document's author or creator are placing the logical "seal of authenticity" on the document; thereby declaring it an official document for you or your organization.

- **Approval Signature:** An approval signature is any signature that was applied without choosing Certify Document. Any signature other than the first one must be an approval signature. Use approval signatures for the following:
 - For any signature other than the first.
 - When you do not need to attest to the document content.
 - When you do not need to restrict what a document recipient can do with the document.
 - When you are approving a document or form for further processing; for example, a purchase order.

6.1.3 Signing User Interface

Signing features are accessible in several ways which vary depending on whether a document already contains signature fields or signatures:

- **Pull down menus:** Pull down menus provide menu items for signing, certifying, and working with signed documents. Items are enabled and disabled based on the current state of the document and what the author has allowed.
- **Right click menus:** For signed documents, right clicking on any signature in the Signatures pane or in the document displays a context menu. Menu items allow you to clear or validate a signature as well as to view the signature's properties.
- **Click on a signature field:** Clicking on a signature field automatically invokes the Sign Document dialog.

6.2 Signing With a Certification Signature

Certifying a document enables the first signer attest to its contents and specify the types of changes permitted for the document to remain certified. Certification helps document authors and recipients determine that documents are legitimate and tamper-proof, thereby enabling trustworthy online transactions and more secure communications.

For example, suppose that a government agency creates a form with signature fields. When the form is complete, the agency certifies the document and allows users to change only form fields and sign the document. Users can fill in the form and sign the document, but if they remove pages or add comments, the document does not retain its certified status. Certifying a document helps ensure that it is not altered without the author's approval.

Certified documents display the following (Figure 63):

- **Blue ribbon icon:** An icon appears next to the digital signature and in the Signature tab.
- **Document restrictions:** The Signature tab displays the certifier-specified restrictions.
- **Explicitly trusted but potentially dangerous content:** A list appears in the Signature tab, if any.
- **Certification attestation:** Depending on user preferences (see , a signer-specified reason for signing may appear in the Signature tab and the signature appearance.

Before certifying, be aware of the following:

- Certification locks certain document elements and limits what a document recipient can do with it.

- Because certification is designed to carry more legal weight than an uncertified document, greater attention to the content and process is typically warranted.
- Certification signatures are automatically validated even if the application preference to automatically validate signatures is turned off.

Document Locking

Certification limits what a recipient can do with a document. Some actions are locked automatically, and some are locked by the certifier. For example, during certification the signer can choose from the following options:

- No changes allowed
- Form fill-in and digital signatures
- Annotations, form fill-in, and digital signatures

General editing, adding or removing pages, and so on are automatically prevented. Any changes that are explicitly locked by the certifier or automatically prohibited by the application invalidate the certifier's signature and revoke the document's certification.

Document Defensibility

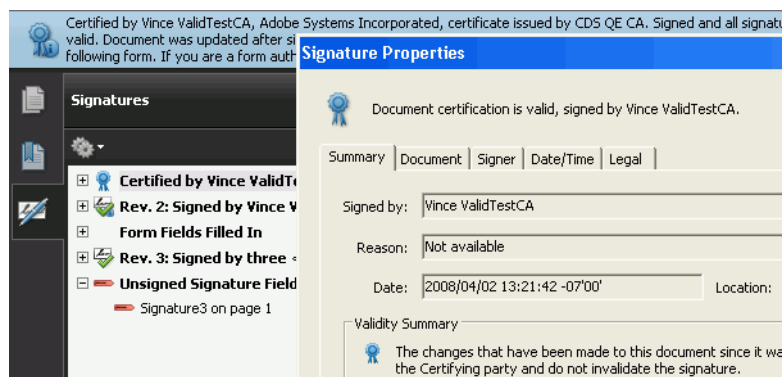
Acrobat has a notion of a document's defensibility which is defined by the features that appear in the legal attestation dictionary, described in Section 8.7.4 of the *PDF Reference* manual. Note that aside from when a signer is certifying, Acrobat does not actively inform the user about the document's legal defensibility.

In any case, a document's legal defensibility improves if it does not contain content that threatens the signer's ability to see what they are signing as well as their ability to certify that what the document recipient sees is the same as that which was certified. Such content includes JavaScript, multimedia, and so on. It is the certifier's responsibility to either remove that content or attest to the fact that such content should be retained.

Hazardous content is revealed to users in two ways:

- Acrobat helps the signer identify such content by scanning the document during the certification signing process. The signer is given the option to embed an attestation in the document about that content that explains why it is present. This behavior is unique to certification signatures and does not apply to approval signatures.
- Document recipients use the **View Document Integrity Properties** button to launch the same content scanning process that was automatically launched when the certifier signed. If the document is certified, the process generates a report that includes the certifier's attestation, if any. Content that has been explicitly trusted by the certifier also appears on the signature tab under **Trusted Content** (Figure 63).

Figure 63 Certified document indicators



Legal Attestations and Warnings Comments

For documents with dynamic content, signer's may want to add a legal attestation or comment indicating the included content has been reviewed is specifically permitted. A legal attestation can only be added on certified documents and during signing. When this option is enabled by the signer's application settings, the Certify Document dialog displays a **Review** button which invokes the PDF Signature Report dialog. The dialog display a **Warnings Comment** field that allows the signer to choose from a default comment or to create a custom comment.

The **Enable Reviewing of Document Warnings** and **Prevent Signing Until Document Warnings Are Reviewed** settings function in tandem and should be set together. Setting both these options to *Always* results in the highest degree of assurance that the signing process is not adversely impacted by malicious content. For details, see ["Setting up the Signing Environment" on page 46](#).

6.2.1 Certification Workflow for Documents with Multiple Signers

Certification allows document authors to define what changes are legal (possible), and it allows the recipient to identify whether a document's problematic features (content that could change the document appearance) originated with the certifier or not. More importantly, this gives the recipient the assurance that if these features in the document are indeed malicious, the certifier can be proven to be at fault. The recommended workflow for defensible signatures can then be described as follows:

1. The document author adds the requisite form fields and any other document customizations. Preventing certain future actions (e.g. to form fill in and signing) can be accomplished ahead of time via JavaScript or during signing.
2. The document is signed with a certification signature. If there is problematic content in the document, the certifier chooses **Review** and adds a Warnings Comment explaining why the content is OK.
3. The document is routed to the next person in the workflow.
4. The document recipient manually validates the certification signature if the application is not set up to validate signatures automatically.
5. Document integrity is verified by right clicking on the certification signature and then choosing **Show Signature Properties > Legal tab > View Document Integrity Properties**. This action invokes the PDF Signature Report dialog which displays a list of problematic content as well as the certifier's comment about that content (if any). For example, a certifier might state why they have added a link to a corporate web site, JavaScript, or some other item.

Note: The certifier's warning comment is not viewable via preview mode.

6. The recipient decides whether or not to continue modifying and signing the document based on the list of warnings and certifier's warning comment (if any).
7. The recipient modifies the document if it is permitted by the certifier (for example, filling in a form).
8. The recipient signs with an approval signature and forwards it to the next recipient (if any).

6.2.2 Setting up a Document for Certification

Authoring Form Fields

When a certified document contains more than one form field, field properties such as locking, placement, naming, tooltips, and even appearance should be specified in ways which help the recipients understand the form and easily determine whether or not data changes have invalidated the signature and certification. For details, see ["Authoring Signable Forms" on page 62](#).

Using Seed Values to Individual Form Fields

6.2.3 You can customize the way a certified document behaves for signers by giving form fields additional features with seed values. For example, you can preconfigure custom signing reasons or limit signing to only those with certificates with predefined characteristics. **Certifying a Document**

Before continuing:

- Configure your application as described in ["Setting up the Signing Environment" on page 46](#).
- Prepare the document for certification as described in ["Setting up a Document for Certification" on page 93](#).

1. Initiate the certification process by doing one of the following:

- Right clicking on a signature field and choosing **Certify with a Visible Signature** ([Figure 67](#)).
- Choosing a menu item:
 - **Advanced > Sign & Certify > Certify with Visible Signature**
 - **Advanced > Sign & Certify > Certify without Visible Signature**

Note: Selecting a field results in signing that field. When a field is not preselected, choosing one of these menu items invokes a dialog which asks you what field you would like to sign.

2. (Optional): If your application is configured to automatically enter preview mode during signing, do the following:

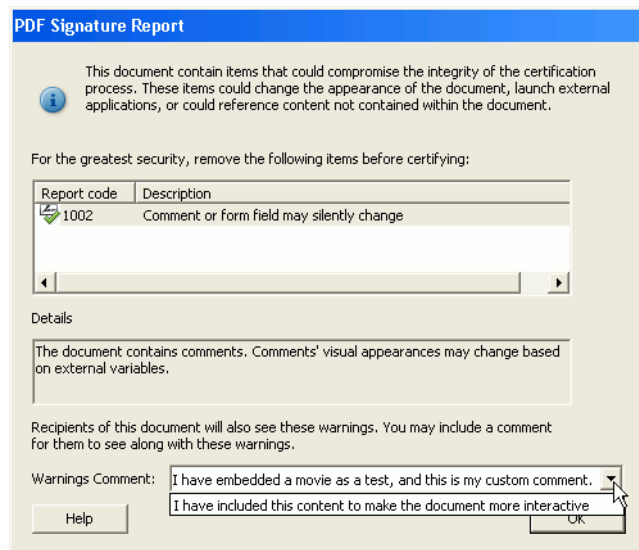
1. Review the text in the Document Message Bar at the top of the document.

2. Choose **View Report** to invoke the PDF Signature Report dialog. Acrobat checks to see if the document contains dynamic content that could adversely impact the integrity of the document. In general:
 - If no errors are listed, the document can always be signed.
 - If only errors with a green check appear, then the document contains rich content that can be suppressed in preview mode. These documents can be signed safely in preview mode.
 - If any error with a red X appears, then the document contains rich content that can not be suppressed in preview mode. The document should not be signed by signers who are concerned that such content may affect how the document appears to the signer or document recipient.
3. Review the warnings and decide whether the document is OK to sign. Choose **Close** when done.
4. Choose **Sign Document**.
5. Configure the Certify Document dialog:
 - **Digital ID:** Select a digital ID. The digital ID selected as the default for signing is automatically selected. For details about changing the default, see [“Specifying Digital ID Usage” on page 15](#).
 - **Password:** Enter a password if the selected digital ID requires it.
 - **Appearance:** Select an appearance or use the default one.
 - **Reason:** If the application is configured to display the **Reason for Signing Document** field, choose an item from the list or enter a new reason.
 - **Location** and **Contact Info** fields: If desired, fill in these optional fields.
6. Set **Permitted Changes after Certifying**:
 - **No changes allowed:** Prevents users, JavaScript, and other hazardous content from changing the document. Since potentially hazardous content is prevented from interacting with the document, that content will not appear in the Signature pane’s Trusted Content list.
 - **Form fill-in and digital signatures:** Limits user interaction to adding data to form fields, including signatures.
 - **Annotations, form fill-in, and digital signatures:** Limits user interaction to adding data to form fields, signing, and commenting.

Tip: If the document contains form fields, specify the settings that make the most sense for the intended workflow. Keep in mind a signature field is a form field. For details, see [“Making a Field a Required Part of a Workflow” on page 63](#).
7. If the **Review** button appears on the dialog (an application setting), choose **Review**. This action invokes the PDF Signature Report again and enables adding a warnings comment or legal attestation (another application setting).

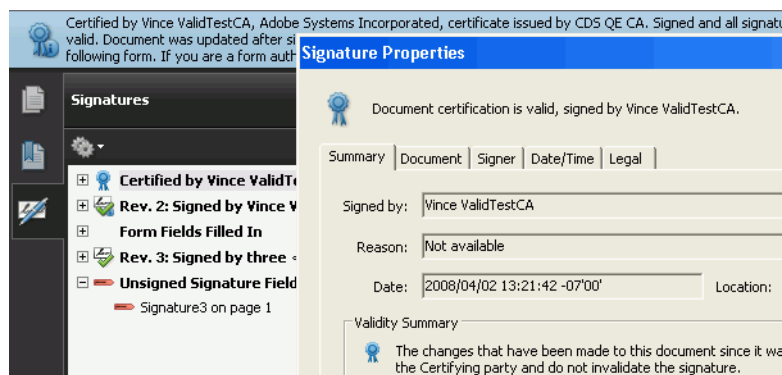
Tip: If you have already reviewed the warnings and don’t need to add a comment, skip this step.

Figure 64 Certifying a document: Document integrity warnings



8. If there are any document warnings in the PDF Signature Report , do the following:
 - Review the warnings and determine whether it is acceptable to certify the document as is. If not remove the problematic content and start over.
 - If the content is ok, enter a **Warnings Comment** for the document recipient. Select the default or enter a custom comment. A comment should tell the reader why the content is there and that you have approved it.
9. Choose **OK**.
10. Choose **Sign**.
11. Save the document.

Figure 65 Certifying a document: Signature



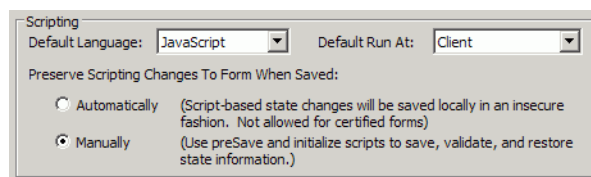
6.2.4 Certifying a Dynamic Form

This information pertains only to LiveCycle Forms Designer which ships with Acrobat.

Because certification is designed to provide a higher level of assurance about document integrity as well as define the boundaries of permitted changes, certifying a dynamic form requires special procedures. Dynamic forms can contain behaviors that prevent certification. Dynamic forms may need to be configured to support certification.

To configure a dynamic form for certifying: Choose **File > Form Properties** and display the Defaults tab. In the Scripting panel, set **Preserve Scripting Changes to Form When Saved** to **Manual**. When the form is subsequently opened in Acrobat or Adobe Reader (with signing rights), certification will be possible.

Figure 66 Dynamic form certification setting



6.2.5 Why Can't I Certify?

In order to certify a document, the certifying signature must be the first one in the document and there must be no restrictions on the document that prevents certifying. Either of these two situations may arise if you are not the author of the document. When a document is ineligible for certification, the certification user interface items are disabled.

In order to certify the document, clear existing signatures, remove the restrictions if you have permission to do so, or save the document under a new name so that you are the document author.

6.3 Signing with an Approval Signature

Documents may be signed with simple approval signatures. When a document is part of a workflow where document elements do not need to be locked, use an approval signature. PDFs can be signed in Acrobat, in Reader (in special cases), or in a browser.

6.3.1 Signing Documents in Acrobat

You can create a new signature field or sign an existing one:

- If your document already contains a signature field, simply click on it and follow the instructions or choose a **Sign Document** menu item.
- If you want to create a new field on-the-fly as part of the signing process, choose a **Place Signature** menu item. You may want to read [Chapter 4, "Authoring Signable Documents"](#) if you would like to control how the document behaves once it is signed.

Before continuing:

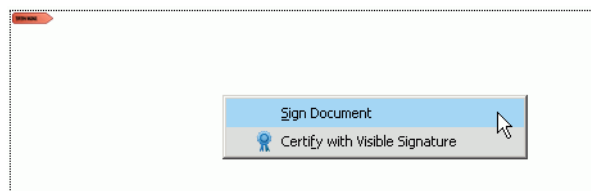
- Configure your application as described in ["Setting up the Signing Environment"](#) on page 46.
- Prepare the document for certification as described in ["Setting up a Document for Certification"](#) on page 93.

To sign a document with an approval signature:

1. Initiate the approval signing process by doing one of the following:
 - Sign an existing field:
 - Click on a signature field.
 - Right click on a signature field and choose **Sign Document** (Figure 67).
2. Sign a new field by choosing **Advanced > Sign & Certify** and then selecting **Approval**.

Tip: Certification signature menu items are disabled if the document has already been signed.

Figure 67 Signature field sign menu



3. (Optional): If your application is configured to automatically enter preview mode during signing (the recommended setting), do the following:
 1. Review the text in the Document Message Bar at the top of the document.
 2. Choose **View Report** to invoke the PDF Signature Report dialog. Acrobat checks to see if the document contains dynamic content that could adversely impact the integrity of the document. In general:
 - If no errors are listed, the document can always be signed.
 - If only errors with a green check appear, then the document contains rich content that can be suppressed in preview mode. These documents can be signed safely in preview mode.
 - If any error with a red X appears, then the document contains rich content that can not be suppressed in preview mode. The document should not be signed by signers who are concerned that such content may affect how the document appears to the signer or document recipient.
 3. Review the warnings and decide whether the document is OK to sign. Choose **Close** when done.
 4. Choose **Sign Document**.

Figure 68 Signing a document: Signature details

Sign Document

Sign As: Mr. Example <example@moosepathbandb.com>

Password: *****

Certificate Issuer: Mr. Example

Info...

Appearance: Standard Text

Mr. Example

Digitally signed by Mr. Example
DN: cn=Mr. Example,
o=moosepathbandb,
ou=moosepathbandb,
email=example@moosepathbandb.com, c=US
Reason: I am approving this document
Location: Fairbanks, Alaska
Date: 2008.08.21 23:38:51 -0700

Additional Signature Information

Reason: I am approving this document

Location: Fairbanks, Alaska

Contact Info: Santa Clause Lane

Click Review to see if document content may affect signing

Review...

Sign Cancel

5. Configure the Sign Document dialog:
 - **Digital ID:** Select a digital ID. The digital ID selected as the default for signing is automatically selected. For details about changing the default, see [“Specifying Digital ID Usage” on page 15](#). If the desired digital ID is not listed, select **Refresh ID List** if your digital ID is on a hardware device which you recently connected, or select **New ID** to create or install a new digital ID now.
 - **Password:** Enter a password if the selected digital ID requires it.
 - **Appearance:** Select an appearance or use the default one.
 - **Reason:** If the application is configured to display the **Reason for Signing Document** field, choose an item from the list or enter a new reason.
 - **Location** and **Contact Info** fields: If desired, fill in these optional fields.
6. Choose **Sign**.
7. Save the document.

6.3.2 Signing in a Browser

To sign a document in a browser, the document must contain an empty signature field:

1. Click on any signature field or choose **Pen Icon > Sign Document** on the Tasks toolbar, and then follow the steps described in [Signing Documents in Acrobat](#).
2. To retain a copy of the signed document, choose the **File > Save A Copy**.

6.3.3 Clearing One or More Signatures

Clearing a signature field deletes the signature but leaves the empty field. Not all signatures can be cleared. You may be prevented from deleting the signature in the following cases:

- You cannot delete someone else's signature.
- If the author of a signature field has marked it to become read-only after it is signed, it can only be cleared by the author.

To clear all signature fields in a document, do one of the following:

- Choose **Advanced > Sign & Certify > Clear All Signatures**.
- In the Signatures tab, choose **Options > Clear All Signatures**.

To clear a single signature field:

1. Right click on a signature.
2. Choose **Clear Signature Field**.

When you receive a signed document, you may want to validate its signature(s) in order to verify who the signer was, when they signed it, and what was actually signed. Depending on how you have configured your application, validation may occur automatically.

However, understanding both how to validate a signature manually as well as what signature components are analyzed during the validation process can facilitate trouble-free workflows and mitigate signature status problems. Participants in signing workflows may also want to configure their environment to streamline the validation process and control what kinds of content in signed documents can be run on their machine.

The following sections provide validation details:

- [“Signature Validity Basics” on page 100](#)
- [“Setting up Your Environment for Signature Validation” on page 102](#)
- [“Validating Signatures Manually” on page 106](#)
- [“Troubleshooting a Signature or Document Status” on page 115](#)
- [“LiveCycle Dynamic Forms and the Warning Triangle” on page 121](#)
- [“Save as 8.1 except that changes to document behavior are detected and invalidate an approval signature: prior versions displayed a yellow triangle upon discovery of changes to document behaviour.” on page 121](#)
- [“Viewing and Comparing Changes and Versions” on page 122](#)

7.1 Signature Validity Basics

As part of the signature validation process, Acrobat and Adobe Reader verify the signer’s identity as well as the document’s integrity.

7.1.1 What Makes a Signature Valid?

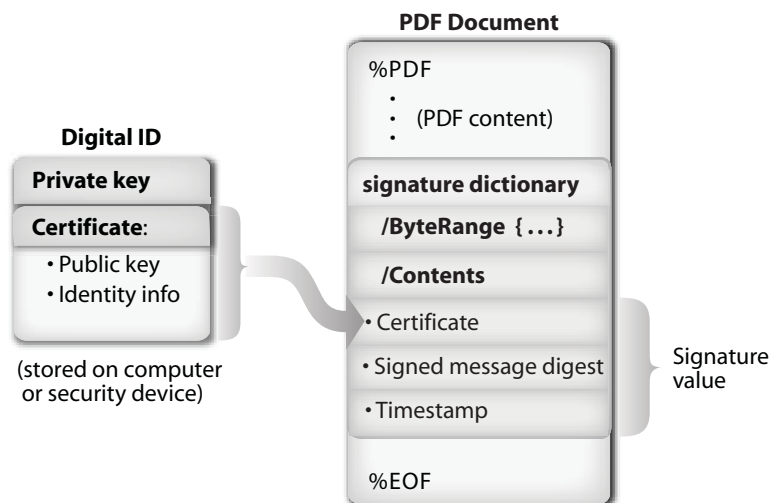
Signature validity is determined by checking the signature’s digital ID certificate status (*is it valid and trusted?*) and document integrity (*has it changed since being signed?*):

- Authenticity verification confirms that the signer’s certificate or that one of its parent certificates exists in the validator’s list of trusted identities and that the signing certificate is valid at that point in time according to the user’s Acrobat or Reader configuration (time of signing, secure timestamp time, or current time).
- Document integrity verification confirms that the signed content hasn’t changed after it was signed or that it has only changed in ways specifically permitted by the signer.

7.1.1.1 Authenticity Verification

Authenticity verification starts with a signer obtaining a digital ID that includes an X.509 certificate. The validator must add that certificate (or have previously added one of its issuing certificates) to their trusted identities list. Either the signer's certificate or one of its issuing certificates must then be explicitly trusted for signing, thereby making it a trust anchor used during signature validation. At validation time, the certificate is processed and analyzed to see if it's valid. That is, Acrobat performs a revocation check and other relevant operations before determining what the signature status will be.

Figure 69 Internal Document Signature components



7.1.1.2 Document Integrity Verification

In signing workflows, document integrity refers to whether or not what was signed has changed after signing in a way that violates any document rules. That is, what the signer signed should be reproducible and viewable on the document recipient's end. At a high level, the Acrobat family of products therefore implements signatures as follows:

- Each signature captures what the document looked like at the signing point in time.
- Only very limited changes are possible after a signature is applied. At most, form field values, additional signatures, and annotations can be changed or added.
- View Signed Version shows exactly what was signed. The signature panel lists post-signing changes.
- A certification signature can tighten the rule to allow less changes than form fields, additional signatures, and annotations.

To verify if a document has changed after signing (has integrity), Acrobat or Adobe Reader must 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.

During the act of signing, the application creates a message digest and then encrypts that digest with the signer's private key. The digest is embedded in the document along with the signature's appearance. Every

time a document is signed, a new digest is created. Thus, each signature is only valid for a specific version of the document.

Because the application stores and numbers a document version for each signature, signature validators can determine what was actually signed. When you validate a signature, a new message digest is created and compared to the digest that was embedded in the document at signing time.

If the two digests are not identical the signature is invalid.

Both signers and signature validators should understand the following about the relationship between signatures and document versions:

- Every time a document is signed, the document's state at the point of signing is stored in the PDF.
- Versions are incrementally numbered beginning with "1."
- A document with 10 signatures will have 10 versions.
- A signature applies to a version (e.g. signature X with version X and signature Y with version Y, etc.).
- When you open a document in Adobe Acrobat or Adobe Reader, the current version always displays.

Note: To learn more about how each signature results in a new version of the document, refer to <http://www.adobe.com/devnet/acrobat/pdfs/DigitalSignaturesInPDF.pdf>.

7.2 Setting up Your Environment for Signature Validation

Document recipients should configure their environment to handle incoming documents in a way that enhances workflow efficiency or meets some business need. While Adobe Acrobat and Adobe Reader provide default options, customizing the environment often provides a better user experience. In large, enterprise environments, your environment may be preconfigured by a system administrator.

Options include the following:

- **Validating Signatures Automatically:** By default, validation occurs automatically. If signatures should not be validated automatically when a document opens, turn this option off.
- **Setting Digital Signature Validation Preferences:** Accept the defaults or configure validation methods such as plugin usage, time display, automatic revocation checking, and other settings.
- **Using Root Certificates in the Windows Certificate Store:** If you would like to trust and use certificates in the Windows Certificate Store for signature validation, turn this option on. Trusting all of these certificates is not recommended.
- **Controlling Multimedia:** When certified documents may contain multimedia, specify whether or not it is allowed to run.
- **Certificate Trust Settings:** Specify whether a certificate should be a trust anchor, trusted for signing, and trusted for certain behaviors in certified documents.

7.2.1 Validating Signatures Automatically

By default, signatures are automatically validated. However, you may want to turn it off for reasons such as:

- You don't care whether the signatures are valid.
- The desktop cannot be configured to validate the signature.

- To gain a small increase in application speed when it opens a document. The difference may be negligible depending on the number of signatures and whether a system administrator has customized revocation checking.

To configure automatic signature validation:

1. Choose **Edit > Preferences** (Windows) or **Acrobat (or Adobe Reader) > Preferences** (Macintosh).
2. Choose **Security** in the left-hand list.
3. Check **Verify signatures when document is opened**.

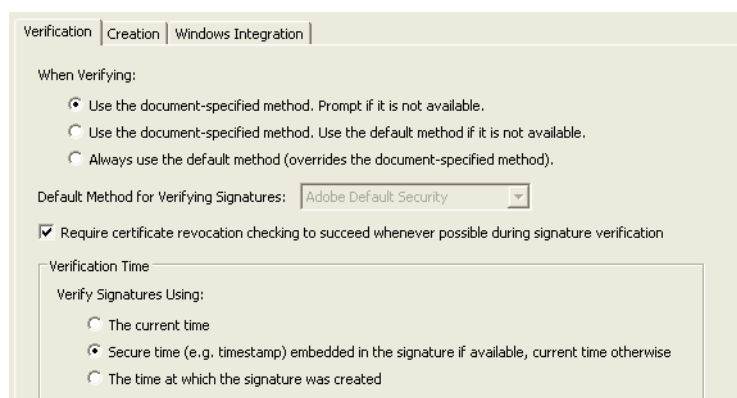
7.2.2 Setting Digital Signature Validation Preferences

To set advanced digital signature preferences:

1. Choose **Advanced Preferences**.
2. Display the Verification tab.

Verification tab options let you specify the validation plugin, default validation methods, whether or not certificate revocation checking is automatic, what time is associated with a validated signature, and whether or not a status icon appears with the signature.

Figure 70 Signature verification preferences



3. Select the signature validation method (use the default setting unless instructed to change it by a system administrator):
 - **Use the document-specified method, prompt if it is not available.**
 - **Use the document-specified method, use the default method if it is not available.**
 - **Always use the default method (overrides the document-specified method).**

Tip: Don't change this setting unless instructed to do so by a system administrator. Signatures are created and validated by plugins. These options specify which plugin is used. Both Acrobat and Adobe Reader provide a default plugin for signing documents and verifying signatures. While the signing and verification usually use the same plugin, this is not always the case. However, a signature always "knows" what plugin is required to verify it.

4. If you have installed a non-default plugin, select your preferred method for verifying signatures.

5. Check or uncheck **Require that certificate revocation checking be done whenever possible during signature validation.**

This option checks certificates against a list of revoked certificates during validation, either with the Online Certificate Status Protocol (OCSP) or the Certificate Revocation List (CRL). If this option is not selected, the revocation status for approval signatures is ignored. *Revocation checking always occurs for certificates associated with certification signatures.*

Note: Signature verification is similar to credit card validation. OCSP checking is like making a phone call to verify the card number. CRL checking is like checking the card numbers against a list.

6. In the Verification Time panel, select a time verification method:
 - **Current time:** The digital signature validation time.
 - **Secure time:** The secure timestamp server time if one is present and trusted, otherwise the current time.
 - **Creation time:** The signature creation time.

7.2.3 Using Root Certificates in the Windows Certificate Store

The Windows Certificate Store contains a store called “Trusted Root Certificate Authorities” that contains numerous root certificates issued by different certification authorities. Certificates are “root” certificates by virtue of being at the top of the certificate chain hierarchy. There are two common ways a certificate ends up in the Windows Certificate Store root directory:

- The computer manufacturer or Microsoft has put them there.
- A company administrator has put them there as part of a company-wide program.

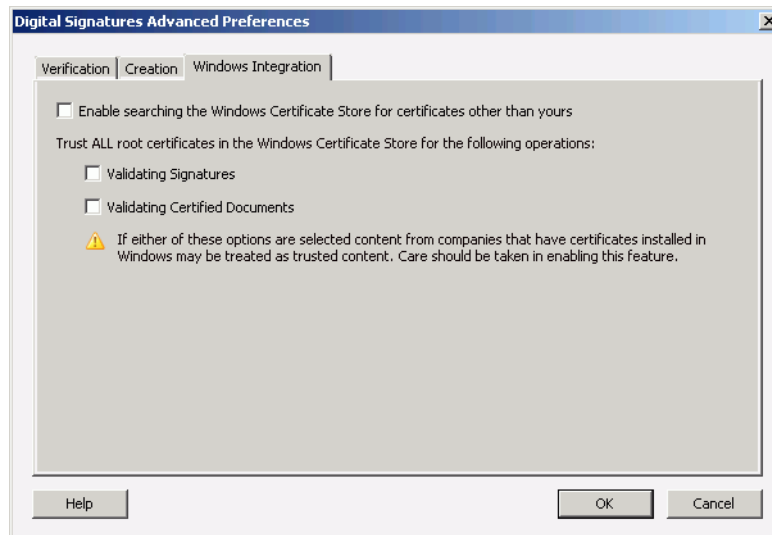
Most home users should not trust all Windows root certificates by default because by trusting a root certificate you may be trusting all the content provided by the company that owns that certificate. Many root certificates ship with Windows, and users may have imported others as a result of some online action.

Enterprise users, on the other hand, should consult company policy to determine whether or not to trust all Windows root certificates for validating signatures or certifying documents. This information should come from an administrator, though your application may already be configured with the correct settings. A common reason to trust Windows roots is so the administrator can manage from a central location the certificates deployed on a network.

To use these certificates for signature validation:

1. Display the Windows Integration tab.

Figure 71 Trusting Windows root certificates



2. Specify the trust level for all root certificates in the Windows Certificates Store:
 - **Validating signatures:** Certificates will be trusted for approval signature validation.
 - **Validating certified documents:** Certificates will be trusted for certification signature validation.
3. Choose **OK**, and exit the preferences dialogs.

7.2.4 Validating Signatures with Timestamps and Certificate Policies

Certificate policies can be used with timestamps, but they can only be verified on the client end, not on the server end. That is, a timestamped signature can not be sent with CRL request with a specific policy OID; however, the client can require that the server response include a specified policy constraint. If the timestamp server returns a response that doesn't include a matching policy OID, then the client would reject the timestamp and it's status would be invalid. The user interface shows the following:

- The signature could be valid, but it's validated at the current time. The Signature pane shows the appropriate icon.
- The timestamp is invalid. The Summary tab of the Signature Properties dialog shows a red X

The require a timestamp to be associated with a particular certificate policy:

1. Configure your application to validate signatures using Secure Time as described in [“Setting Digital Signature Validation Preferences” on page 103](#).
2. Configure a policy constraint for a trust anchor in your trusted identities list:

Note: xxxx: question: Am I choosing the certificate for the timestamp server I have previously added to my trusted identities list?

1. Choose **Advanced > Managed Trusted Identities**.
2. In the **Display** drop down list, choose Certificates.
3. Select the timestamp server's certificate that will be used as a trust anchor.

4. Choose **Edit Trust**.
5. Choose the Policy Restrictions tab.
6. Enter a certificate policy OID.
7. Choose OK.

Note: If the timestamp server returns a response with a policy not specified by the client, the timestamp signature will be invalid due to an invalid policy constraint.

7.3 Validating Signatures Manually

Unless the application is configured to do otherwise, signatures are validated automatically when a document opens. If they are not validated or if a signature needs to be revalidated, you can validate one or more signatures manually.

Validating a signature allows you to verify the signer's identity and determine whether the displayed document is identical to what was signed (or that only allowed changes were made):

- Identity verification confirms the signer's certificate or one of its parent certificates exists in the list of trusted identities and is not expired or revoked.
- Document integrity verification confirms that the signed content hasn't changed since signing or that it has only changed in ways specifically permitted by the signer. Signatures can be validated one at a time or all at once.

Before validating a signature, it is a good idea to understand what a signature is and how signature status is indicated. For details, see the following:

- ["What Makes a Signature Valid?" on page 100](#)
- ["Status Icons and Their Meaning" on page 113](#)

7.3.1 Validating Signatures with Adobe Reader

The process for validating one or more signatures in Adobe Reader is similar to Acrobat. However, the top level menu item is labelled **Document** instead of **Advanced**. Therefore, the validation paths are as follows:

- **Document > Sign > Validate All Signatures**
- Click on a signature in the document or the Signatures pane, right click, and choose **Validate Signature**.

7.3.2 Validating a Single Signature in Acrobat

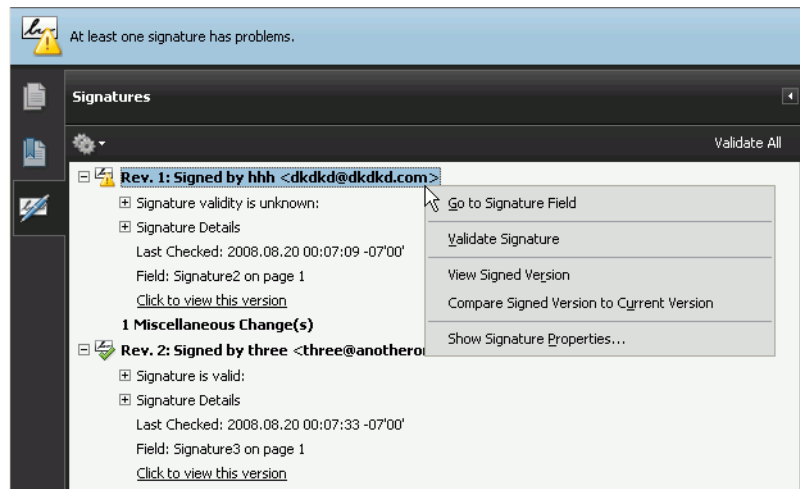
Signatures can be validated one at a time or all together as described in ["Validating All Signatures in Acrobat" on page 107](#). Signature validity can be determined by viewing its associated icon. A green check mark indicates the signature is valid without reservations. Other icons indicate there may be a problem.

There are several ways to verify a signature manually:

- Right click on any signature in the Signatures pane or in the document, and choose **Validate Signature**.

- Highlight a signature in the Signatures tab, and choose **Advanced > Sign & Certify > Validate Signature** or open the Signature Properties dialog and choose **Validate Signature**.

Figure 72 Signatures tab: Validate signature



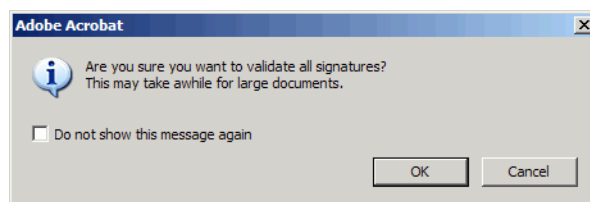
7.3.3 Validating All Signatures in Acrobat

All signatures in a document may be validated simultaneously. This feature is particularly useful if the auto-validate option has been turned off.

To validate all signatures:

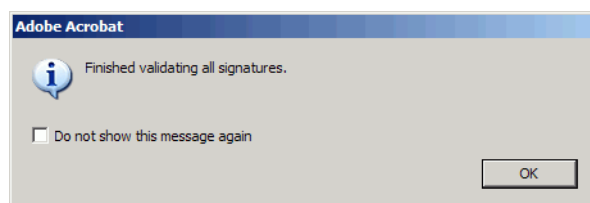
1. Choose **Advanced > Sign & Certify > Validate All Signatures**.
2. If a dialog appears asking if all signatures should be validated, choose **OK**. This dialog does not appear if you have previously checked **Do not show this message again**.

Figure 73 Validate all signatures dialog



3. If a dialog appears confirming all signatures have been validated, choose **OK**. This dialog does not appear if you have previously checked **Do not show this message again**.

Figure 74 Signature validation confirmation



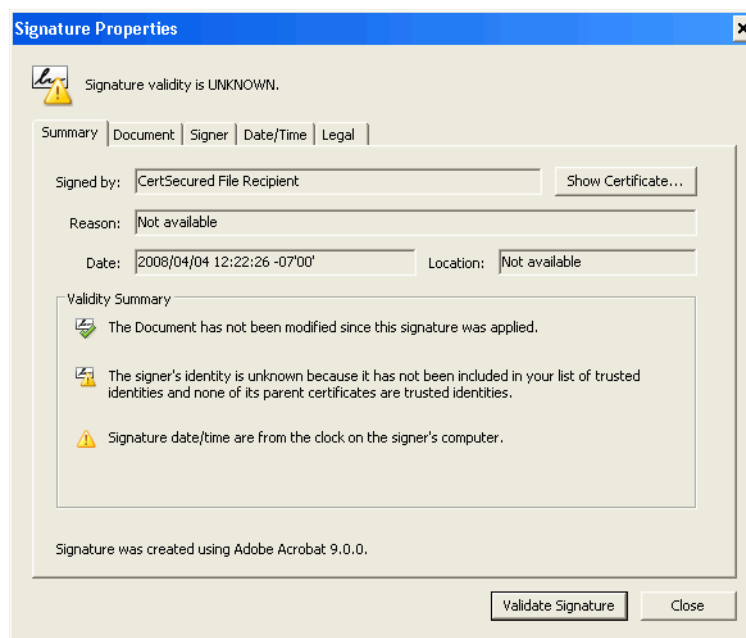
7.3.4 Validating an Problematic Signature (trusting a signer on-the-fly)

If a signer's digital ID certificate has not been explicitly trusted, the signer is untrusted and the signer's signature validity will be *problematic*. When a signer has not been trusted ahead of time, you can trust their certificate for signing and certifying directly from the signature. After their ID (contact information and certificate) is added to your list of trusted identities, the signature can be validated.

To add an someone's certificate a list of trusted identities:

1. Display the Signature Properties dialog by right clicking on any signature in the document or the Signatures tab and choosing **Show Signature Properties**.
2. Choose the Summary tab (Figure 75).

Figure 75 Signature Properties: Summary



3. Choose **Show Certificate**.

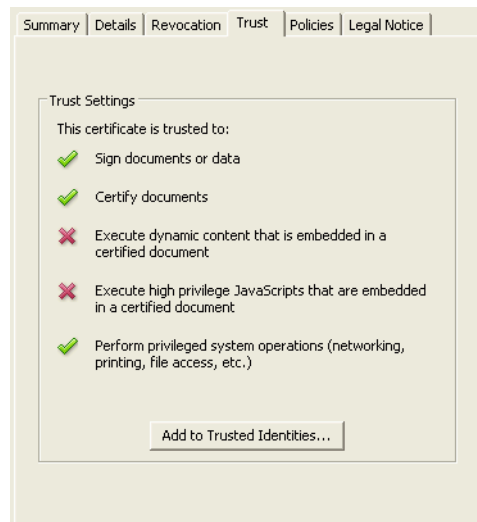
Adding an unverified digital ID certificate to the trusted identity list could pose a security threat. This is particularly true for self-signed IDs that are not issued by a third-party certificate authority. For details, see [“Verifying the Identity of Self-Signed Certificates” on page 118](#).

4. When the Certificate Viewer appears, choose the Trust tab (Figure 76).
5. Choose an item in the left-hand certificate path field. There may be one or more certificates which make up a certificate chain.

Tip: If the bottom-most certificate on the chain is selected, then only that certificate will be trusted. If the top-most certificate is selected, then any certificates having that certificate as a root will be trusted. For example, if the root certificate is issued by VeriSign and it is trusted, then other certificates having VeriSign's certificate as the root (also issued by them) will also be trusted. It is a best practice to trust the topmost certificate that you are

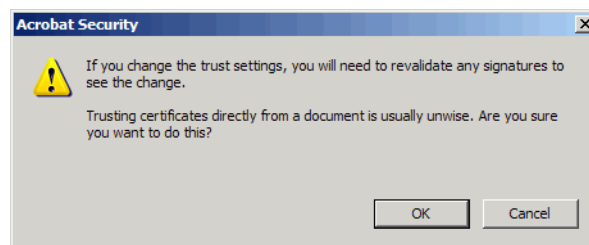
willing to trust. Revocation checking starts at the bottom of a chain (begins with the end entity), and once it reaches a trusted root revocation checking stops.

Figure 76 Certificate viewer: Trust tab



6. Choose **Add to Trusted Identities**.
7. When asked if the certificate should be trusted, choose **OK**.

Figure 77 Trusting certificate from a document warning



8. When the Import Contact Settings dialog appears, configure its trust levels. For details, see ["Certificate Trust Settings" on page 35](#).
The Policy Restrictions tab will not appear if there are no policies associated with this certificate.
9. Choose **OK**.
10. Choose **OK**.
11. Choose **Close**.
12. Right click on the signature and choose **Validate Signature**.

Tip: The yellow triangle icon on the signature will not change until the signature is revalidated.

7.3.5 Validating Signatures for other Document Versions

When you open a document, the latest version is always displayed. You can see whether the signature associated with earlier signed versions of the document are valid simply by opening the Signature pane and viewing the status icon and text.

Documents with multiple signatures contain the elements needed to reconstruct any previous version of itself as it existed at the time of signing. In other words, Acrobat and Adobe Reader “remembers” that version A is signed, that changes were made to version B, and so on. Therefore, it may be necessary to view the signed version in order to see what content was actually signed. Viewing the signed version allows you to check if the signature is valid for a particular document version.

To view the signed version of a document.

1. Right click on the signature you want to validate in the document or in the Signatures tab.
2. Choose **View Signed Version**. The application opens the signed version of the document.
3. Revalidate the signature if necessary.

Tip: For more about versioning, see [“Document Integrity Verification” on page 101](#).

7.3.6 Validating Signature Timestamps

If you know a signature is timestamped or your workflow requires timestamps, read the following sections. At a high level, the rules are as follows:

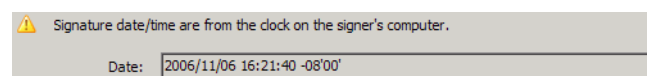
- You can configure Acrobat to use timestamps by setting the verification preferences as described in [“Setting Digital Signature Validation Preferences” on page 103](#).
- If set, the secure timestamp server time is used if one is present and trusted, otherwise the current time is used.
- Timestamp validity does not affect signature validity. A signature can be valid even if the timestamp server’s certificate is invalid or expired.
- The signature validation time appears in the Date/Time tab of the Signature Properties dialog.

Local Time versus Timestamp Time

Signature times tell you that a document and signature existed prior to the indicated time. All signatures are associated with the signer machine’s local time, but they may also include a timestamp time if the signer’s application was configured to use a timestamp server. Because users can set their machine time forward or back, local time is less reliable than a timestamp time. Local times are labelled as such in the Date/Time and Summary tabs of the Signature Property dialog ([Figure 78](#)).

Note: Because signature appearances only display local time, the appearance time will be different from the timestamp time shown in the Date/Time tab of the Signature Properties dialog.

Figure 78 Timestamps: Local, machine time

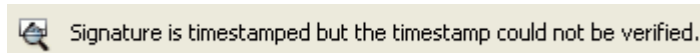


What is a timestamp?

A timestamp is like a signature inside of a signature. Like signatures, timestamps are provided by someone (a timestamp authority) who uses a certificate to confirm their identity. A timestamp's certificate must be valid (not revoked by the issuer) and trusted (by you) for the timestamp to be valid. Timestamp certificate status appears in the Date/Time and Summary tabs of the Signature Property dialog:

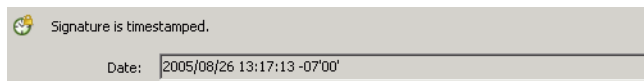
- Untrusted timestamp certificates appear as follows:

Figure 79 Timestamps: Untrusted stamp



- Trusted timestamps that have been added to the Trusted Identities list and have been explicitly trusted for signing appear as follows:

Figure 80 Timestamps: Trusted stamp



How Do I Validate a Timestamp in a Signature?

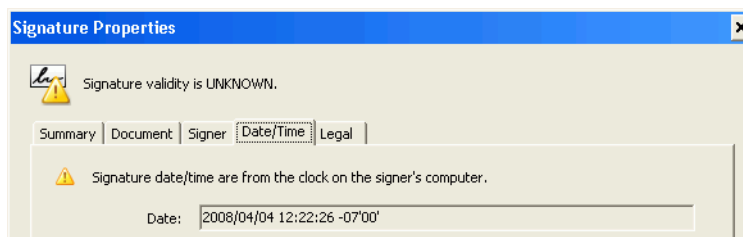
Validating a timestamp is the process by which you check to see if the timestamp was applied and that its certificate is valid. In order to validate a timestamp, you need to manually verify:

- **The timestamp was applied:** If a timestamp fails for some reason (the server cannot be found, the network is down, etc.), timestamping fails silently.
- **The timestamp certificate is trusted:** If a timestamp was applied, the certificate must be trusted by adding to your trusted identities list.

To verify that a signature has been properly timestamped:

1. Display the Signature Properties dialog by right clicking on any signature in the document or the Signatures tab and choosing **Show Signature Properties**.
2. Choose the Date/Time tab. Timestamp status is indicated by the icon and associated text:
 - **Warning triangle:** Timestamping failed or a timestamp is not present and the local time is used. Verify the signer used a timestamp.
 - **Magnifying glass:** A timestamp may have been used but you have not yet trusted the timestamp certificate. Proceed to the next step.
 - **Clock:** A timestamp with a trusted certificate was used.

Figure 81 Timestamps: Date/Time tab



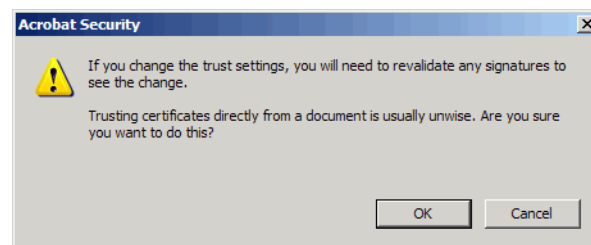
Note: The following steps add a timestamp certificate to your list of trusted identities.

3. Choose **Show Certificate**.
4. When the Certificate Viewer appears, choose the Trust tab.
5. Choose an item in the left-hand certificate path field. There may be one or more certificates which make up a certificate chain.

Tip: If the bottom-most certificate on the chain is selected, then only that certificate will be trusted. If the top-most certificate is selected, then any certificates having that certificate as a root will be trusted. For example, if the root certificate is issued by VeriSign and it is trusted, then other certificates having VeriSign's certificate as the root (also issued by them) will also be trusted. It is a best practice to trust the topmost certificate that you are willing to trust. Revocation checking starts at the bottom of a chain (begins with the end entity), and once it reaches a trusted root revocation checking stops.

6. Choose **Add to Trusted Identities**.
7. When asked if the certificate should be trusted from within the document, choose **OK**.

Figure 82 Revalidate signatures warning



8. When the Import Contact Settings dialog appears, configure the its trust levels. For details, see ["Certificate Trust Settings" on page 35](#).
The Policy Restrictions tab will not appear if there are no policies associated with this certificate.
9. Choose **OK**.
10. Choose **OK**.
11. Choose **Validate Signature** on the Date/Time tab of the Signature Properties dialog. The icon should change to a clock if the check is successful.

7.3.6.1 When Timestamps Can't be Verified...

If a signature is timestamped but cannot be verified, open the Trusted Identity Manager and verify:

- A certificate is associated with the timestamp server. Verify the timestamp authority's certificate is in the certificate list.
- The trust level of the certificate is set. Choose a certificate and verify that the trust level is set for signing. The certificate must either be a trust anchor or be issued by someone whose certificate you have specified as a trust anchor.

7.4 Status Icons and Their Meaning

By default, signatures are validated automatically when a document opens. You can change this behavior as described in [“Validating Signatures Automatically” on page 102](#). Signature and document status’ are represented by status icons and text both in the document, on the Signatures pane, on the Document Message Bar, and in the Signature Properties dialog, and elsewhere.

Note: For a higher level of assurance, do not rely solely upon the visual inspection of status icons. Review the Signature Properties dialog for revocation and trust information as well as the signer’s certificate details.

7.4.1 Signature Status Definitions

To determine a signature’s status, the application checks the signature’s digital ID certificate status (is it valid) and document integrity (has it changed since being signed).

The rules for determining signature status are as follows:

- **Valid** signatures used a valid and trusted certificate and the document has not changed or has changed in ways specifically permitted by the author.
- **Problematic** signatures are associated with certificates that cannot be validated or lack a trust relationship with the signer.
- **Unknown** signatures indicate that the signature validity state has not been checked.
- **Invalid** signatures either have an invalid certificate or the document has changed in ways specifically prohibited by the author.

7.4.2 Document Status Definitions

In addition to the individual status for each signature, the Document Message Bar displays the document’s overall status. The document status is essentially a summation or “rollup” of all the signature status’ AND the effect of document changes after the last signature.

For example, a form might have two valid signatures and a valid document status. However, when someone types into the form’s text box, both the signature status’ get the blue “i” information icon indicating that something has changed after the signatures were added. The document message bar now shows a yellow triangle indicating that there are unsigned changes in the document. If the form is signed again, the overall document status changes back to valid as indicated by the green check.

The rules for determining a document’s status are as follows:

- If there is only one signature and the document hasn’t changed since it was signed, then the document status is identical to the signature status.
- The status is flagged as problematic if there are unsigned changes following the last approval signature.
- The status is unknown (magnifying glass) if the authenticity verification check could not complete.
- Like a signature status, if either the authenticity verification or document integrity check fails, the overall document status is invalid (red x).

7.4.2.1 Signature status cheat sheet

Document status appears in the Document Message Bar.

Document status appears in the Document Message Bar.

Signature status appears in the Signature pane.

Signature status appears in the Signature pane.

Document status appears in the Document Message Bar.

Signature status appears in the Signature pane.

Signature status appears in the Signature pane.

Signature status appears in the Signature pane.

Document status appears in the Document Message Bar.

Signature status appears in the Signature pane.

Signature status appears in the Signature pane.

Signature status appears in the Signature pane.

Doc Status	Identity Check	Document integrity check
Certified	Verified for all signers. First signature is a certification signature.	Document has not changed or only contains permitted changes. ¹
Certified	Verified for all signers. First signature is a certification signature.	Re-signed X times, ¹ then permitted changes were made, then re-signed X times. ¹
Certified	Verified for certifier, but one or more subsequent signers may have a problem.	Unsigned (permitted) changes after an approval signature or some signatures problematic. ¹
Valid	Verified for all signers.	Document has not changed or only contains permitted changes. ¹
Valid, changes	Verified for all signers.	Permitted changes were made, it was then re-signed with one or more approval signatures. ¹
Problematic	Unverified. Certificate validation problem. ^{3,4,5,6}	Unsigned changes or a problem with one or more of the signatures. ^{1,2}
Unknown	Identity check has not executed. ⁷	Integrity check has not executed. ⁷
Invalid	Signer's certificate was bad, expired, or revoked at the time of signing. ⁸	Illegal changes made, document corrupted, or policy restrictions violated. ^{1,9}

Sig Status

Sig Status	Identity Check	Document integrity check
Certified	Verified. Certification signature.	Document has not changed or only contains permitted changes. ¹
Valid	Verified. Approval signature.	Document has not changed.
Valid, changes	Verified. Approval signature.	Permitted but signed changes exist. ¹
Problematic	Certificate validation problem. ^{3,4,5,6}	Unsigned changes after this signature. ^{1,2}
Unknown	Check has not executed. ⁷	Integrity check has not executed. ⁷
Invalid	Signer's certificate expired, or revoked at the time of signing. ⁸	Illegal changes made, document corrupted, or policy restrictions violated. ^{1,9}

ACROBAT 9.0 TROUBLESHOOTING GUIDE:

- 1 View change history in Signature pane. View Signed Version to see what was signed.
- 2 Sign changes or review and accept them (and ignore the warning).
- 3 Verify Internet connection, verify server is running (if possible).
- 4 Signature expired. Check app's validation time preference or have signer re-sign.
- 5 Review certificate's validity, revocation info, and associated policies.
- 6 Set a trust anchor, or request a new certificate from signer and set its trust level.
- 7 Save document. Turn on automatic validation or manually validate signature.
- 8 Have the signer resign with a valid certificate.
- 9 Have the document re-signed, check policy restrictions and security of your workflow.

STATUS DEPENDS ON TWO CHECKS:

Signer's Identity: Verifies the signer's certificate is trusted (in the validator's list of trusted identities) and valid at the time specified by the AcrobatReader configuration: signing time, timestamp time, or current time.

Document integrity: Verifies the signed content hasn't changed or that it has only changed in ways permitted by the signer.

THERE ARE TWO TYPES OF SIGNATURES:

Certification: Certifies the document. Only one allowed per document and it must be the first one. Can lock the document or specify allowed actions such as signing, form fill in, and commenting.

Approval: Signs but doesn't certify. Any number allowed.

7.5 Troubleshooting a Signature or Document Status

Note: In enterprise settings, a system administrator may have configured your application to behave differently than described below.

Ideally, signature validation should result in the display of a green check icon for approval signatures or a blue ribbon icon for certification signatures. Icons always appear in the Document Message Bar, the Signature Pane, and in the Signature Properties dialog.

In addition to the signature and document status icons and text, key tools for troubleshooting signatures include the following:

- **Signatures pane:** Displays all of the signatures, status, change history, and links to signed versions.
- **Signature Properties dialog:** The dialog provides five tabs that display signature information and buttons for performing document validation tasks. It also provides a **Show Certificate** button for invoking the Certificate Viewer.
- **Certificate Viewer:** The viewer provides certificate-specific information and buttons for performing certificate validation tasks. Together, the Signature Properties dialog and Certificate Viewer should provide you with enough information to either successfully validate a signature or reject the document as insecure.

7.5.1 Troubleshooting an Identity Problem

If the signature status or overall document status indicates that there is a problem with verifying the authenticity of the signer, you may need to verify the signer and/or decide whether to trust that signer.

Note: Trust does not happen automatically. For a signature to be trusted, your application must be configured for that trust. That configuration could be the result of actions by Adobe, your administrator, or you.

To troubleshoot authenticity problems, open the signature panel and expand the information for the problematic signature. Read what it says the problem is, and then take one or more of the following actions:

1. If the status is unknown (displays a magnifying glass) and the icon shows a magnifying glass, it is possible signature validation did not occur.
 - Validate the signature(s) as described in [“Validating Signatures Manually” on page 106](#).
 - Verify you have an internet connection and the application is configured properly.
 - Since the problem may not be with your application, try again later.
2. If the status is problematic (displays a warning triangle), do the following:
 - Verify the signer is in your trusted identity list and that you have configured their certificate or one of their certificate’s issuing certificates as a trust anchor.
 - Verify you have trusted the signer’s certificate for signing and (if necessary) certifying. The Certificate Viewer’s Trust tab indicates the certificates trust level (right click on a Signature, choose

- Show Signature Properties** and then **Show Certificate**). Specify the certificate's trust settings as described in ["Certificate Trust Settings" on page 35](#).
- Verify that a revocation check occurred. Open the Certificate Viewer's Revocation tab (right click on a Signature, choose **Show Signature Properties** and then **Show Certificate**). Check the following:
 - If revocation checking occurred, **Problems encountered** is active and you can select the button to view a description of the problems.
 - If revocation checking did not occur at all, **Check revocation** is active and you can select the button to check revocation manually.
 - If online revocation checking is required, it may have failed as a result of no online access or an application problem.
 - 3. If the status is invalid (displays a red X), the signer's certificate is invalid. Do the following:
 - Contact the signer. The signer may need to get a new digital ID and re-sign a new document.
 - Policy restrictions on a trust anchor can result in signature invalidity. If you have set a policy restriction, determine if that is the problem remove the restriction.
 - 4. If you still cannot pinpoint the problem, or you need help with some of the steps above, read the following:
 - ["Troubleshooting Digital ID Certificates" on page 116](#)
 - ["Displaying the Signer's Certificate" on page 117](#)
 - ["Verifying the Identity of Self-Signed Certificates" on page 118](#)
 - ["Checking Certificate Revocation Status" on page 119](#)
 - ["Exporting a Certificate Other than Yours to a File" on page 120](#)

7.5.1.1 Troubleshooting Digital ID Certificates

Someone becomes your trusted identity when you import their valid digital ID certificate and set a specific trust level for that certificate. You can set trust levels ahead of time if you have access to those certificates. If you do not have access to those certificates, simply validate and trust certificates "on-the-fly" as you receive individual documents. As shown in [Table 12](#), the Certificate Viewer provides six tabs with functionality for working with and verifying digital ID certificates.

Table 12 Certificate Viewer information

Tab	What it shows	What you can do
Summary	Signer and Issuer information, validity dates, and intended usage.	Export the certificate to a file.
Details	Certificate data such as subject, issuer, used algorithms, public key, and so on.	The data can be used in a variety of ways such as using the digests to verify the certificate's origin.
Revocation	Shows certificate validity status of a revocation check and provides an explanation.	<p>Signer Details: Open the certificate in the Certificate Viewer. The button is only active if the revocation check was successfully completed.</p> <p>Problems encountered: View revocation checking problems. The button is only active if revocation checking occurred but failed.</p> <p>Check revocation: Enables manual revocation checking. The button is only active if no checking occurred AND a check is possible.</p>
Trust	Lists the user-specified certificate trust settings.	Add the certificate to the Trusted Identity list.

Table 12 Certificate Viewer information

Tab	What it shows	What you can do
Policies	Lists policy OIDs associated with this certificate, if any. Describes the policy.	View policy details.
Legal Notice	Displays a generic legal disclaimer, the certificate issuer's policy statement, issuer notice, and link to the policy, if any.	If an issuer policy is used, the policy can be displayed.

7.5.1.2 Displaying the Signer's Certificate

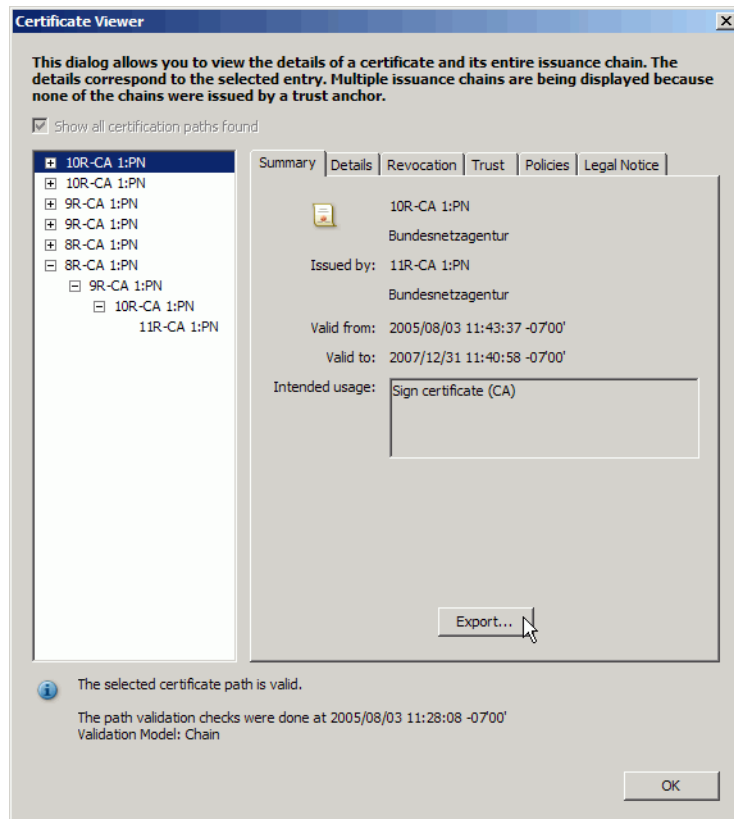
When a certificate is displayed in the certificate viewer, you can check certificate validity, trust settings, associated policies, and other details that help you establish the owner's identity. The Certificate Viewer provides six tabs that displays certificate data and allows you to manage that certificate (Table 12).

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Manage Trusted Identities**.
2. Choose **Certificates** in the **Display** drop-down list.

Note: In addition to this method, you can also display the certificate from any signature or certificate security method workflow where a **Show Certificate** or **Certificate Details** button appears.

3. Select the certificate.
4. Choose **Show Certificate**. The Certificate Viewer displays the certificate. (Figure 84). The following details are available:
 - **Left hand panel:** The certificate chain.
 - **Bottom area:** A description of the certificate, path validity statement, path validation time, and sometimes the type of validation.
 - **Summary tab:** Owner, issuer, validity period, intended usage. An **Export** button allow users to export the certificate to a file.
 - **Details tab:** Lists all the certificate fields (extensions) and their values.
 - **Revocation tab:** Indicates whether a revocation check occurred and the result. Allows users to initiate a manual check and analyze problems.
 - **Trust tab:** Displays the certificates trust level. Provides an **Add to Trusted Identities** button that allows the user to add the certificate to the trusted identity list and set its trust level.
 - **Policies tab:** Displays policy restriction information for a signature to be valid, if any.
 - **Legal Notice tab:** Displays other certificate policies and a button which links to that policy, if any.

Figure 83 Certificate Viewer



7.5.1.3 Verifying the Identity of Self-Signed Certificates

Certificates are usually issued by a trusted, third-party certificate authority such as VeriSign. However, anyone can set up a certificate authority or create a self-signed certificate purporting to be anyone else. For self-signed certificates or those issued by unknown or untrusted certificate authorities, it is prudent to verify the certificate owner's identity before being added to them trusted identity list.

To verify the origin of the certificate:

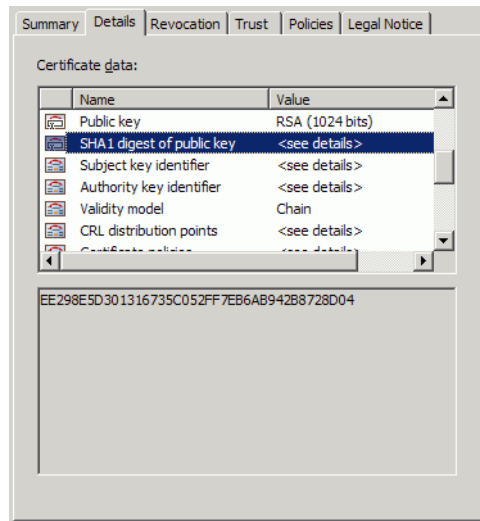
1. Display the certificate in the Certificate Viewer:

- If the certificate is embedded in a signature, right click on the signature, choose **Show Signature Properties**, display the Summary tab, and choose **Show Certificate**.
- If the certificate is in an FDF file, double-click the attached file, and choose **Certificate Details** in the Import Contact Settings dialog.

Tip: Double clicking a file other than an FDF will likely install the certificate in the Windows Certificate Store. If the file is .cer, .p7b, or some other format and you want to import into the Acrobat certificate store, save the file locally and import it into the Trusted Identity Manager as described in ["Importing Someone's Certificate" on page 166](#)

2. Display the Details tab.

Figure 84 Certificates: Verifying originator



3. In the Certificate data panel, scroll to MD5-digest and SHA-1 digest, and note the numbers.
4. Contact the certificate's originator, and verify the MD5-digest and SHA-1 numbers are correct.
5. After the certificate is verified, display the Trust tab and add the certificate to the trusted identities list.
6. Specify certificate trust settings so that it can be used as a trusted root, to certify documents, and so on. For details, see ["Certificate Trust Settings" on page 35](#).

7.5.1.4 Checking Certificate Revocation Status

Only the certificate issuer (a certificate authority) has the right to revoke a certificate. A certificate could be revoked because its security might be compromised, it could be invalid for some reason, or the owner of the ID might have left the company. Adobe applications check revocation status as part of its public key authentication.

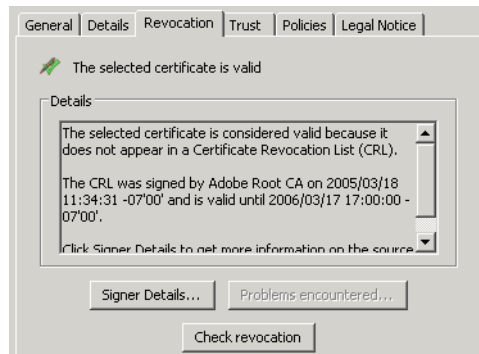
To check a certificate's revocation status:

1. Choose the Certificate Viewer's Revocation tab.
2. Choose **Check Revocation**.

The button is only active if revocation checking has not occurred and it is possible to do a check. Revocation checking is not possible for trusted roots and self-signed certificates.

3. Choose **OK**.

Figure 85 Trusted Identities: Viewing revocation status



7.5.1.5 Exporting a Certificate Other than Yours to a File

Users in enterprise settings can send problem certificates to their system administrator or help personnel for troubleshooting. Certificates may be exported from the Trusted Identity Manager, or from the Certificate Viewer.

To do export a certificate from the Trusted Identity Manager:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Manage Trusted Identities**.
2. Choose **Certificates** in the **Display** drop-down list.
3. Select the certificate.
4. Choose **Show Certificate**. The Certificate Viewer displays the certificate. (Figure 84).
5. Choose **Export**.
6. Email the certificate or save it to a file as described in [“Exporting Your Certificate” on page 158](#).

7.5.2 Troubleshooting a Document Integrity Problem

If the signature status or overall document status indicates that there could be a document integrity problem, you need to figure out if and how the document changed during the workflow.

. To check document integrity:

1. If the status is unknown and the icon shows a magnifying glass, it is possible signature validation did not occur.
 - Validate the signature(s) as described in [“Validating Signatures Manually” on page 106](#).
 - Verify you have an internet connection and the application is configured properly.
 - Since the problem may not be with your application, try again later.
2. If the status is problematic (displays a warning triangle) shows a, the document has changed but those changes are legal because they have been allowed by the document author. To determine what changed, do any of the following:
 - Open the Signature pane and expand the signature tree to view the change history,

- Right click on a signature and choose **View Signed Version** or choose **Click to view this version** in the Signature pane to view the version that signed. Review the status for this version. For details, see [Chapter 8, "Document Integrity and Preview Mode"](#).
 - Open the **Form Fields Filled In, or Annotations Created or Modified** item to see which fields or annotations were changed or added.
 - Review the document changes as described in ["Viewing a List of Post-Signing Modifications" on page 122](#).
 - Perform a page by page and line by line comparison of the problem version with any of the signed versions. For details, see ["Comparing a Signed Version to the Current Version" on page 123](#).
3. Based on the discovered document changes, determine whether you should trust the document.
 4. If the status is invalid (displays a red X), illegal changes have been made to the document, there is no way to undo those changes without further changing the document illegally. Do the following:
 - Contact the sender to resolve the problem. Secure the workflow so that illegal changes are prevented.
 - Policy restrictions on a trust anchor can result in signature invalidity. If you have set a policy restriction, determine if that is the problem remove the restriction.
 5. If you still cannot pinpoint the problem, or you need help with some of the steps above, read the following:
 - ["Troubleshooting Digital ID Certificates" on page 116](#)

7.5.2.1 LiveCycle Dynamic Forms and the Warning Triangle

Document Integrity Checks for 8.0

Acrobat 8.0 considers all scripts executed during document construction to potentially modify the document even if it's designed with a read-only query or some other "no change" action. The presence and detection of those scripts would trigger Acrobat to display the yellow warning triangle, thereby indicating that the document may have changed. Scripts that invalidate certification would be prevented from executing so it would not be possible for such scripts to invalidate certifying signatures.

Document Integrity Checks for 8.1

Acrobat 8.1 does not consider all scripts executed during document construction to potentially modify the document, and the detection of a script does not cause Acrobat to flag the document as changed. The application compares the digitally signed and current document versions to determine if the current version has been modified. If there is a change, then a warning triangle appears on the approval signature. For example, the following changes are detected:

- Changes to the value of an LiveCycle form field (including clearing it).
- Changes to the properties of an LiveCycle form field.

Document Integrity Checks for 9.0

- Save as 8.1 except that changes to document behavior are detected and invalidate an approval signature: prior versions displayed a yellow triangle upon discovery of changes to document behaviour.

7.5.2.2 Viewing and Comparing Changes and Versions

Document authors and recipients often need to know if a document has changed since it was signed. Acrobat keeps track of a document's version number, stores previous document versions in their entirety, and enables users to compare document versions by work and page. When you open a document, the latest version is always displayed whether or not it is the signed version. Document recipients should always remember the following:

- Every time a document is signed, the entire document at the point of signing is stored in the PDF as an incrementally numbered version.
- A signature signs a document at a specific point in time; that is, signature X signs version X and signature Y signs version Y. You can view exactly what the document looked like at that point in time using **View Signed Version**.

For details, see the following:

- [Chapter 8, "Document Integrity and Preview Mode"](#).
- ["Viewing a List of Post-Signing Modifications" on page 122.](#)
- ["Comparing a Signed Version to the Current Version" on page 123.](#)

7.5.2.3 Viewing a List of Post-Signing Modifications

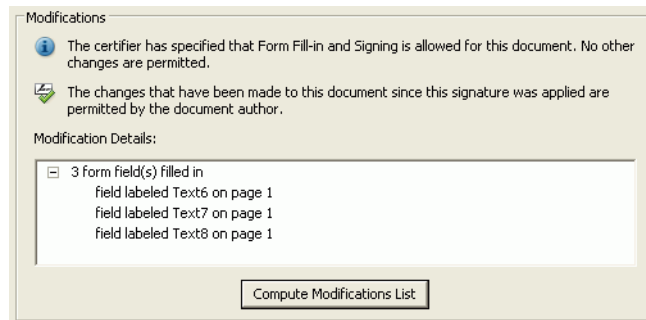
Because it is possible to change a document without changing its appearance, the list of post-signing modifications is often a superset of what is visually displayed when comparing documents using Document Compare. Therefore, a thorough analysis of a signed document's integrity includes viewing the document modifications list.

To view a list of post-signing modifications, open the Signature pane and view the change history. All changes are listed in chronological order.

Note that you can also do the following to get a more condensed list:

1. Right click on a signature and choose **Show Signature Properties**.
2. Choose the Document tab.
3. Choose **Compute Modifications List**. The list is capable of showing the following:
 - Fields (with field names) that have been created, modified, deleted, or filled in.
 - Annotations (comments) that have been created, modified or deleted.
 - Pages that were created, modified or deleted.
 - Spawned pages, deleted spawned pages, and modified spawned pages.
 - Attachments that have been added, deleted, or modified.
 - Miscellaneous: Some changes which occur in memory or cannot be explicitly listed are labelled miscellaneous.

Figure 86 Digital Signature Properties: Modifications panel



7.5.2.4 Comparing a Signed Version to the Current Version

Note: The Compare feature is not available in Adobe Reader.

As you revise a document and save it to a different name or location, you can verify that you have the latest version by comparing it against an older version. If you're revising a document using comments you received during a review, you may need to view a previous version to make sure that you included all the revisions. As a reviewer, you may want to check the updated document against an older version to make sure that the author has incorporated all of your requested changes.

Document Compare does not compare comments or other annotations in the document.

To automatically compare a signed document version with the current view:

1. Right click on a signature and choose **Compare Signed Version to Current Version**.
2. When the two versions appear side by side, review the highlighted areas to review what was changed.

This method compares the two versions page by page. Compare completes by opening a temporary document that summarizes the differences. The first two pages summarize the changed, added, deleted, or moved pages, taking document A as the original and document B as the modified version (Figure 88).

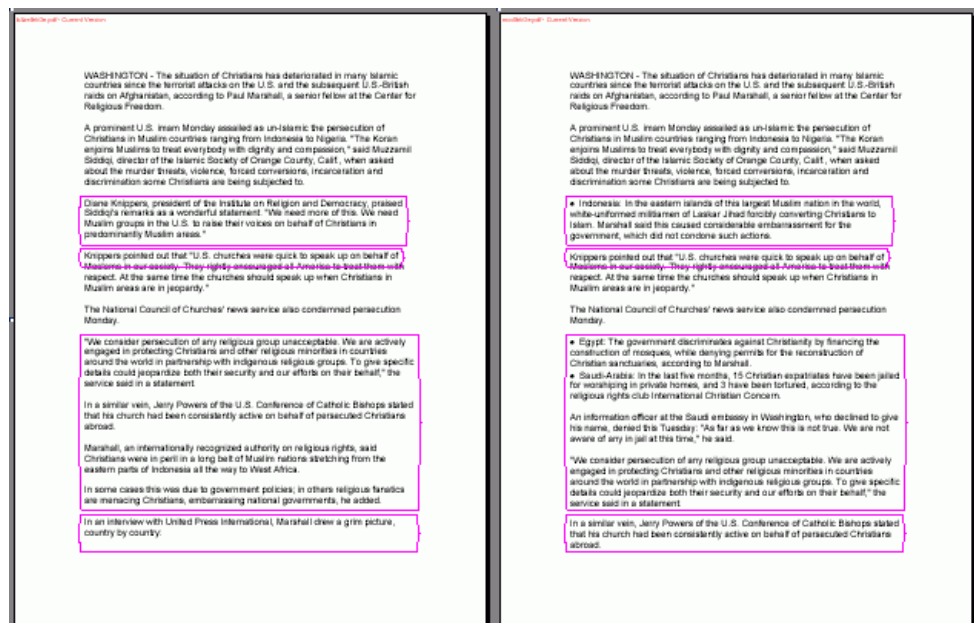
The differences are displayed as follows:

- Even numbered pages (the pages on the left on the two page document view) are pages from document A.
- Odd numbered pages (the pages on the right on the two page document view) are pages from document B.
- Pages that were moved are not shown in the report.
- Any added page, which only exists in document B, is paired with a blank page in the report. Naturally, the added page will be on the right, and the blank page will be on the left in the two-page view.
- Any deleted page, which only exists in document B, is paired with a blank page in the report. Naturally, the deleted page will be on the left, and the blank page will be on the right in the two-page view.
- Pages that were in both documents but were modified are paired with each other in the report. There will be purple hexagons around regions in the two documents that were modified.
- Pages that were not modified will not be in the report.

Figure 87 Compare: By page summary report



Figure 88 Compare: By page



7.6 Document Behavior After Signing

A document's behavior will likely change after it has been signed. Some of it's content may not work (multimedia may not play), some of the application's menu items may be disabled so that you can't use them, and so on. How a document behaves on your desktop could be the result of one or more factors:

- How the document was authored. Were restrictions or requirements placed on the signature fields?
- How a document was signed. Was an approval or certification signature used? Did the signer place restrictions placed on future permissible actions?
- How is your environment configured? Have you trusted the signer's certificate for certain actions? Do you use preview mode?

Note: These items interact in complex ways. In many cases, it's better to accept the application defaults unless instructed to change them by someone knowledgeable about Acrobat's security features.

7.6.1 JavaScript and Dynamic Content Won't Run

High privilege JavaScript and dynamic content in documents will only run if you have explicitly trusted the sender's digital ID certificate for such actions. Because scripts and dynamic content represent a security risk, Acrobat prevents some of those operations by default. For details, see ["Certificate Trust Settings" on page 35](#).

7.6.2 Certifying a Document is Prevented

Only one certification signature is allowed in a document; therefore, it must be the first one.

7.6.3 Form Field Fill in, Signing, and/or Other Actions Don't Work

Once a document is signed or certified, at most only form fill-in, additional signatures, and addition of annotations is allowed. All other operations on the document are disabled as they would invalidate the signature. A certified document may impose further restrictions.

Locking the document after signing also reduces the allowed actions on a document to almost none.

Since 8.1, Acrobat has defined PDF features that should be avoided when producing a document that has a deterministic and repeatable visual rendering. Acrobat's preview mode feature is designed to display that rendering to users during signing and signature validation.

Preview mode analyzes a document for signing best practices and generates a report and messages which indicate the presence of content that might violate those practices. Preview mode suppresses content that may alter the document's appearance, thereby allowing you to view, sign, and validate the document in a static and secure state.

In general, documents that contain no dynamic content (and only recognizable PDF content) are safer to sign than documents with content that can impair one's ability to see what they are signing and validation. At a high (and simplistic) level, static documents are good; dynamic documents are risky.

To mitigate the risk associated with signatures in complex documents, preview mode makes the document as static as possible and also generates a report about what behaviors could and could not be suppressed. Using this feature involves the following:

- Learning when and how to use preview mode during signing or signature validation.
- If preview mode or warnings are reviewed, analyzing the result to determine if a document should be trusted.

Tip: There is only a loose correlation between signature or document status and the information displayed in the PDF Signature report. The report is simply a tool designed to identify the presence of potentially malicious dynamic content that could affect the integrity of your signing workflow. Therefore, signature status and PDF signature report information may appear to be contradictory.

8.1 Preview Mode and Signing Workflows

While preview mode may not be needed in your workflow, using it offers a higher degree of assurance that signers and signature validators are viewing the same document and that the signed version is unlikely to change in signing workflows. Users should decide for themselves whether to sign and even trust signatures in documents that contain dynamic content. Therefore, preview mode is recommended for both signing and validation.

Preview mode can be invoked during any part of a workflow:

- **Before or during signing:** When the preference **View documents in preview document mode when signing** is turned on, preview mode is automatically invoked.
- **After signing:** Right click on any signature and choose **View Signed Version**.

Application behavior varies slightly depending on how preview mode was invoked. In general, however, PDF signature report information can be viewed on the document message bar (DMB) text or by choosing the **View Report** button on the DMB.

It may be prudent to not trust a document that generates errors for content that could not be suppressed. For other errors, you may wish to analyze the document for non-conforming content in order to evaluate whether you should trust the document.

8.2 Preview Mode and Validation (View Signed Version)

Acrobat and Adobe Reader store in signed documents a unique document version for every signature in the document. In other words, they “remember” that version A is signed, that changes were made to version B, and so on. When you open a document, the latest version is always displayed. However, it is sometimes useful to view the signed version in order to see what content was actually signed.

To view the signed version of a document.

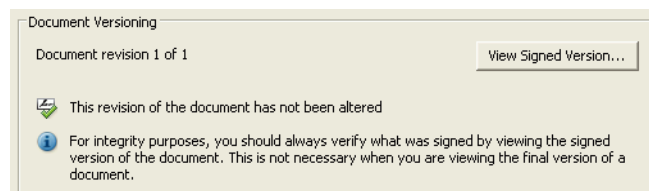
1. Open the Signatures pane, and verify the signature status.

2. Right click on a signature and choose **View Signed Version**.

View Signed Version is essentially a rollback feature that enables the signature validator to view the document version as it was at the point in time when it was signed.

3. Choose **View Report** in the Document Message Bar to view a report about the dynamic content that could and could not be suppressed (if any).
4. Right click on the signature and choose Show Signature Properties for more detail about the signature.

Figure 90 Digital Signature Properties: Document Versioning panel



8.3 PDF Signature Reports

Signature workflows often require a document that has a deterministic and repeatable visual rendering that is consistent with the state of the document as signed. This variant does not concern itself with content in a document that is not rendered or does not affect rendering. Examples of such content include metadata and bookmarks.

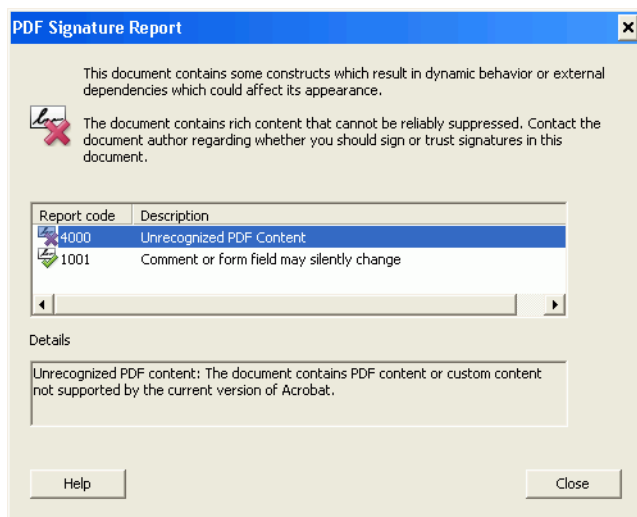
The level of document’s conformance to signing best practices is determined by the presence or absence of certain content. There are three categories:

- “Content preview mode cannot suppress” on page 128
- “Content preview mode can suppress” on page 128
- “No external dependencies or dynamic content” on page 129

Content preview mode cannot suppress

Documents that contain content or behaviors which are dynamic or invisible and which cannot be suppressed in preview mode automatically invoke the PDF Signature Report dialog. For example, preview mode cannot suppress (eliminate from the document or make static) externally referenced images, multimedia content outside of the PDF file, and TrueType fonts. Such content is associated with a red X. Users can highlight an error to see more detail (Figure 91). The signer should decide whether or not to trust the document for signing. If you are not the author, contact the author for additional information.

Figure 91 PDF Signature Report: Content which cannot be suppressed in preview mode



Content preview mode can suppress

Preview mode can suppress certain types of dynamic content. When documents are signed in preview mode, the signed view of the document (with the dynamic content suppressed) is saved so that signature validators can use the View Signed Version feature to validate the signature and see what the signer saw when they were signing.

The **View Report** button opens a dialog that lists discovered rich content. Suppressed content is associated with a green check (Figure 93). If you are concerned about the presence of rich content even though it is suppressed in preview mode, review the error codes and descriptions in the PDF Signature Report dialog for more information.

Figure 92 Document Message Bar: Suppressible rich content

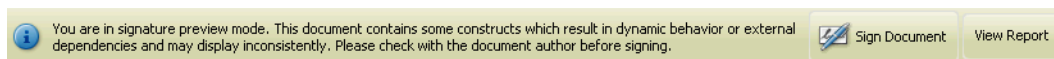
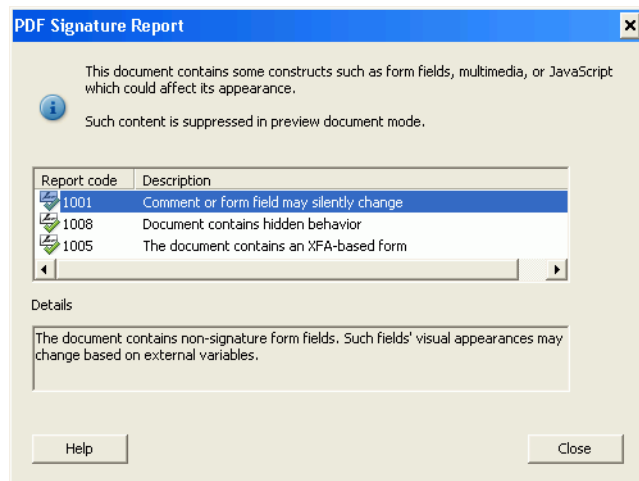


Figure 93 PDF Signature Report: Suppressed content



No external dependencies or dynamic content

For the highest level of document integrity insurance, do not allow dynamic content or any content with external dependencies. These documents are the safest to sign because they can be reliably displayed and do not require a special viewer or preview mode to be signed or to later view what was signed.

Figure 94 Document Message Bar: No dynamic content message



8.4 Signature Report Error Codes

As shown in the following tables, report errors categorize content as one of the following:

- **Dynamic features:** Presentations, user-launched multimedia, JavaScript, dynamic forms, and so on.
- **PDF content with variable rendering:** JavaScript, non-embedded fonts, and so on.
- **External content:** Hyperlinks, alternate images, linked files, and so on.
- **Uncategorized content:** Unrecognized or malformed PDF content.

Table 13 dynamic feature warnings

String	Code	Description
Document contains hidden behavior	1000	The document contains hidden actions that may not be intended or known by the end user. Actions include JavaScript actions (document open, save, etc.), playing multimedia, executing a menu item, and so on.
Comment or form field may silently change	1001	The document contains non-signature form fields. Such fields' visual appearances may change based on external variables.
Comment or form field may silently change	1002	The document contains comments. Comments' visual appearances may change based on external variables.

Table 13 dynamic feature warnings

String	Code	Description
Document may silently launch menu items	1003	The document contains named actions that may launch menu items without the user's knowledge.
Presentation elements may change appearance	1004	Presentations are not allowed since a presentation may contain animations or other elements that may change document appearance or behavior.
The document contains a dynamic form	1005	XFA-based (dynamic forms) documents are not allowed since such forms could alter the document's appearance or behavior.
Document contains links to external PDFs	1006	The document links to external PDFs on the Internet, file system, or network and it has no control over the nature of that linked content. Embedded Go-To actions must not refer to external hierarchies.
Comment or form field may silently change	1007	Disallowed annot type: <annot type>. One or more form fields are associated with a 3D object, file attachment, multimedia, or other dynamic objects.
Document contains hidden behavior	1008	Disallowed action type: <action type>. The document contains hidden actions that may not be intended or known by the end user. Actions include JavaScript actions (document open, save, etc.), playing multimedia, executing a menu item, and so on.
Document contains hidden behavior	1009	The document's content is divided into layers that can be silently displayed or hidden on the fly.

Table 14 PDF Content with variable rendering

String	Code	Description
Page content may silently change	2004	Visual elements may change based on external variables. For example, a logo may change color based on time or zoom level. No postscript XObjects allowed.
Document may not open in the future	2006	Some or all of the content is encrypted and the encryption method is not available in standard Acrobat installations. For example, the document may be protected by the Adobe Policy Server. Document contain streams encrypted using crypt filter.
Page content may silently change	2007	The document author has enabled image interpolation. No image interpolation is allowed.
Page content may silently change	2009	The document uses a PDF transfer function that interprets and replaces color. For example, it could replace black with white. Extended graphic state should not use the TR key
Page content may silently change	2010	The document uses a PDF transfer function that interprets and replaces color. For example, it could replace black with white. If present, the extended graphic state's TR2 key must be set to default
Page content may silently change	2011	The document's extended graphic state uses the FL key. The key is a number that indicates how much flatness tolerance should exist when drawing objects. Content may display differently from Acrobat to other applications
Page content may silently change	2012	Image XObject must not contain an alternate version
Text appearance may silently change	2013	Document contains non-embedded fonts. When the document opens on a system that does not have the requisite fonts, Acrobat will replace them with some other font. Users should always turn on font-related warnings. The non-embedded fonts warning is turned off by default. It can be turned on by setting the DigSig\bEnNonEmbFontLegPDFWarn preference to true. The disallowed font type warning is also turned off by default and can be turned on by setting the DigSig\bTrueTypeFontPDFSigQWarn preference to true. These are Windows registry or Mac plist settings. See the <i>Security Administration Guide</i> for more details.
Text appearance may silently change	2014	Disallowed font type: . True type and TrueType-based OpenType fonts are not allowed because they are programs and may change the document's appearance based on external variables.

Table 15 External Content

String	Code	Description
Document links to external content	3000	Document links to images not in the PDF. No external XObjects allowed.
Document links to external content	3001	Document links to images not in the PDF that are used as alternates. For example, an alternate, high resolution images might be specified for printing. Images must not contain an OPI alternate version.
Document links to external content	3002	Document contains external streams. The author has flagged some PDF bytes as a stream which may get data from an external source.
Document links to external content	3003	Document links to images not in the PDF that are used as alternates. For example, an alternate, high resolution images might be specified for printing. Form XObject must not contain an OPI alternate version.

Table 16 Uncategorized warnings

String	Code	Description
Unrecognized PDF content	4000	Unrecognized PDF content: The document contains PDF content or custom content not supported by the current version of Acrobat. The document may have been created by a later version of Acrobat.
Page content may silently change	4001	Unrecognized drawing operator: The document contains PDF content or custom content not supported by the current version of Acrobat. The document may have been created by a later version of Acrobat.
PDF content contains errors	4002	Malformed drawing instructions: Syntax error. Page content violates the grammar for page content definition. For example, the instruction might specify drawing a square but the syntax for doing it is incorrect.

External Content and Document Security

Document access to internal and external content such as the Internet, attachments, and embedded multimedia represents a security risk. Users should configure their application so that it operates at an acceptable risk level. In enterprise settings, administrators should either preconfigure client installations or distribute instructions for setting up the application correctly.

For details about application settings that control how documents interact with elements outside of the document, see the following:

- [“Enhanced Security” on page 132](#)
- [“Controlling Multimedia” on page 136](#)
- [“Setting JavaScript Options” on page 139](#)
- [“Working with Attachments” on page 141](#)
- [“Controlling Access to Referenced Files and XObjects” on page 145](#)
- [“Internet URL Access” on page 146](#)

9.1 Enhanced Security

Like all other file formats, a PDF or an FDF file could contain a malicious script or perform some action that can damage a computer or steal data when it is run. Enhanced Security enables control of potentially risky behavior by allowing users to turn on enhanced security and either prevent dangerous actions altogether or else only permit them based on whether they reside in a privileged location. These behaviors include: silent printing; cross-domain access, external stream access, and internet access; and script and data injection. For example, if a PDF from your company has an embedded script, it downloads; otherwise, it is blocked.

Acrobat and Reader provide two ways to block potentially unsafe PDFs:

- A system administrator can add Internet domain names to the `crossdomain.xml` file on the server. Only files from locations listed in the `crossdomain.xml` file can be downloaded to individual computers.
- Individuals can identify specific files, folders, or URLs (hosts) as privileged locations in the Enhanced Security Preferences dialog. Items in privileged locations bypass enhanced security restrictions. Any actions, such as loading data from the Internet or running a script are allowed. For example, Enhanced Security blocks FDFs from loading data from unknown websites. If you add the FDF to your list of privileged locations, Acrobat allows the data to be loaded.

At a high level, Enhanced Security includes the following:

- Preventing silent printing; cross-domain access, external stream access, and internet access; and script and data injection.
- Allowing dangerous behavior for only the specified privileged locations. These locations can be a file, directory, or host server.
- FDF behavior is fundamentally altered when this feature is on. For details, see *Distributing and Migrating Security Settings*.

- Enhanced Security interacts with the Trust Manager so that the most permissive setting takes precedence. For example, if a document is signed with a certificate that's trusted for an action that Enhanced Security would normally prevent, that action will be allowed.

9.1.1 Enabling Enhanced Security

This feature is only available to Acrobat users. Enhanced security is designed to limit document behaviors in workflows where those behaviors are perceived as a vulnerability or security risk. In workflows where security is critical, users can turn this feature on. Administrators can also configure the application for enhanced security and lock down the setting so it can't be changed by the user.

- Control of the following behaviors at the file, folder, and host level:
 - **Cross-domain data access:** Different origin data downloads (from where the current PDF resides) to the PDF.
 - **Data injection:** Injection of the data into the document; most notably into form fields via JavaScript or into the document and application via FDF files. Note that FDF files are used specifically to transport data and that turning on enhanced security will impair FDF's ability to do that.
 - **External streams access:** Access to external XObjects; that is, references to objects such as images that reside outside the PDF.
 - **Script injection:** Injection and execution of JavaScript.
 - **Silent printing:** Printing initiated by JavaScript or other document-based methods without user interaction. This includes printing to a file on disk and is not restricted to hardware printers.
 - **Weblink:** Connecting to the web.

To turn on Enhanced Security and specify privileged locations, do the following:

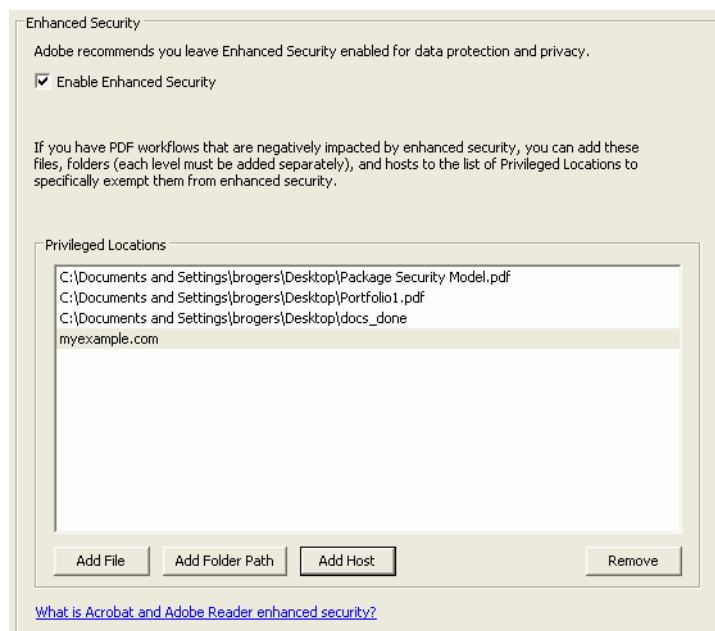
1. Choose **Edit > Preferences > Security (Enhanced)**.
2. Check the Enable Enhanced Security checkbox.
3. (Optional) Set a privileged location by selecting one or more of the following buttons:

Note: Privileged locations bypass Enhanced Security restrictions.

- **Add File:** Select this option if you only have one or two files from a location that you want to add. If you have a large number of files you know you can trust, it can be more practical to place them into one privileged folder.
- **Add Folder Path:** If you have a large number of files that you trust, specify an entire directory.
- **Add Host:** Enter the name of the root URL only. For example, enter `www.adobe.com` but not `www.adobe.com/products`. To only allow higher privileges for files accessed from secure connections, select the option for **Secure Connections Only (https:)**.

4. Choose **OK**.

Figure 95 Enhanced security: Configuration dialog



9.1.2 Changes in FDF Behavior

FDF files are data exchange files. Like .acrobatsecurity files, they help you move certificate, server, and other data from one machine to another. This data transfer usually involves some mechanism such as data injection into a PDF form field, installing files, executing a script, and so on. These actions represent a potential security risk, and in some environments that risk may be unacceptable. Enhanced Security disables some FDF functionality unless those FDF files originate from a specifically privileged file, folder, or server. [Table 17](#) lists the high level rules defining FDF behavior.

Tip: If you need to configure your environment for enhanced security or need to troubleshoot FDF workflows that may not be working as expected, see [“Enhanced Security” on page 132](#).

Table 17 Rules for opening a PDF via FDF

Action	FDF location	PDF location	8.x behavior	9.x behavior
Opening a target PDF	local	local	PDF opens and no authentication required.	Same.
Opening a target PDF	local	http server	PDF opens	User authorization required unless trusted via enhanced security feature.
Opening a target PDF	https server	http server	PDF opens and no authentication required.	Same.
Opening a target PDF	https server	local	Blocked	Http hosted FDFs cannot open local files.

Table 17 Rules for opening a PDF via FDF

Action	FDF location	PDF location	8.x behavior	9.x behavior
Data injection	n/a	n/a	Allowed	Allowed if: <ul style="list-style-type: none"> • Data returned via a form submit with url#FDF. • FDF has no /FDF key. • cross-domain policy permits it.
Data injection	server	browser	Allowed	Allowed if: <ul style="list-style-type: none"> • Link to PDF contains #FDF=url. • FDF has no /FDF key. • x-domain policy permits it.
Data injection	server	Acrobat/Reader	Allowed	Allowed if: <ul style="list-style-type: none"> • PDF makes EFS POST/GET and FDF sends data in https response to same PDF. • x-domain policy permits it.
Data injection	Varied	Varied	Allowed	Authorization required if enhanced security is on and document is not set as a privileged location.
Script injection	Any	Any	Allowed	Injection is blocked unless if enhanced security is on and FDF is not in a privileged location.

Examples of Prevented Behavior

The following are examples of disallowed actions when Enhanced Security is on:

- If the PDF opens in the browser, and the URL to the PDF contains a #FDF=url, then the FDF data specified by that url may be injected into the open PDF if the FDF has no /F key and if the PDF may receive data from the FDF based on the cross domain policy.
- If the PDF opens in the Acrobat/Reader standalone application and the FDF data comes back in the https response to a POST/GET initiated by the PDF, then the FDF data may be injected into the open PDF if the PDF specified in the FDF is the PDF that made the POST/GET and if the PDF may receive data from the FDF based on the crossdomain policy (i.e. * in crossdomain.xml).

Examples of Allowed Behavior

The following are examples of scenarios where FDF data injection does need a user-authorization dialog when Enhanced Security is on:

- You submit data from a PDF in the browser and the URL has #FDF at the end. The FDF that comes back has an /F key pointing to a different PDF which needs to get loaded (everything is happening in the browser). The FDF data gets injected into the second PDF.
- Same as above, except it all happens in the Acrobat rather than in the browser. In this case, the #FDF at the end of the URL is not needed.
- The "spontaneous FDF" case: In the browser, an unsolicited FDF arrives (via a link from an HTML page before and Acrobat is not running yet), and the FDF has an /F key for a PDF that it needs to open and populate.
- Opening a link of the form <http://A.com/file.pdf#FDF=http://B.com/getFDF>.

9.1.3 Interaction with Trust Manager

Enhanced Security interacts with the Trust Manager so that the most permissive setting takes precedence. For example, if a document is signed with a certificate that's trusted for an action that Enhanced Security would normally prevent, that action will be allowed.

9.1.4 Make Privileged Folder Locations Recursive

You can extend privileged locations to be recursive by configuring the registry a reg setting. For details, refer to the *Security Administration Guide for Acrobat 9.0 and Adobe Reader 9.0*.

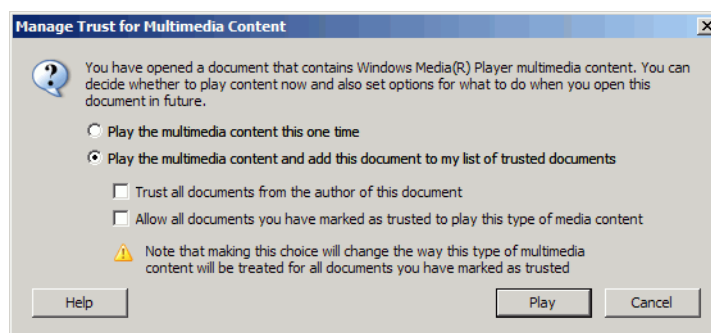
9.2 Controlling Multimedia

The Acrobat family of products have a notion of *trusted documents* and *other documents* (documents that have not been trusted). For the purposes of multimedia playback, every document will exist in one category or the other. For this reason there are two sets of trust options in the Multimedia Trust panel--one for documents that are trusted and one for documents that are not. In order to understand multimedia behavior then, you need to know whether or not a document is trusted so that you can determine which set of options for multimedia playback will be used.

There are two ways a document can become trusted:

- It can be signed with a valid certification signature, and you have trusted the signer's certificate for dynamic content.
- If your multimedia trust preferences result in a prompt asking whether you want to play multimedia, the Manage Trust for Multimedia Content dialog will offer various options that may allow you to trust the document.

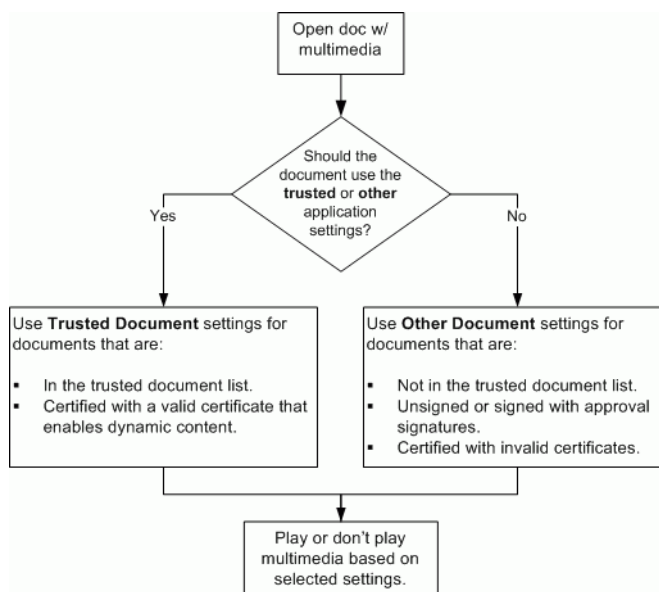
Figure 96 Manage Trust for Multimedia Content dialog



Once a document is trusted, it is added to the Trusted Document list and will always use the preferences set for trusted documents. You can clear this list by selecting **Clear** in the Multimedia Trust panel (Figure 98).

Caution: Membership on the trusted document list is permanent until the list is manually cleared. Therefore, once a document is on that list, changing the certificate trust level to disallowing dynamic content will have no effect.

Figure 97 Multimedia behavior workflow



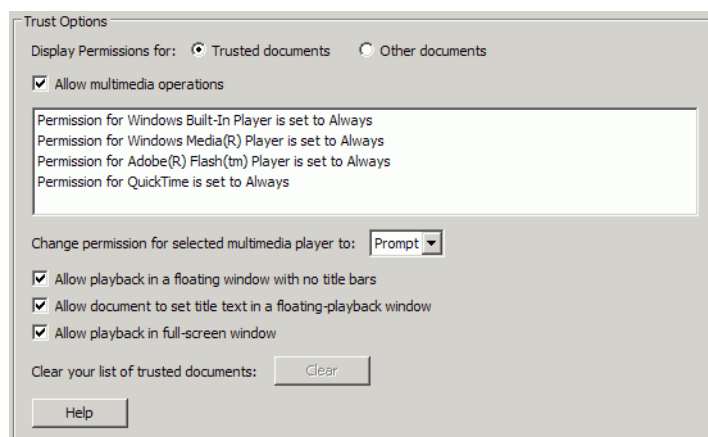
9.2.1 Configuring Multimedia Trust Preferences

Controlling multimedia behavior in documents begins with specifying preferences for *trusted documents* and *other documents*.

To configure multimedia preferences:

1. Open the Multimedia Trust Manager:
 - Acrobat and Adobe Reader (Windows): **Edit > Preferences > Multimedia Trust**
 - Acrobat and Adobe Reader (Macintosh): **(Application) > Preferences > Multimedia Trust**

Figure 98 Multimedia Trust (legacy)



2. From the **Display Permissions for** radio buttons, choose **Trusted documents** or **Non-trusted documents**. The Trust Manager displays the selected trust preferences (Figure 98).
3. Configure the Trust Options panel:

1. Check or uncheck **Allow multimedia operations**.
 2. Set multimedia player permissions as follows: Select the player in the list and select an option from the **Change permission for selected multimedia player to** drop-down list:
 - **Always:** The player is used without prompting.
 - **Never:** Prevents the player from being used.
 - **Prompt:** Prompts the user to enable the player when a media clip tries to use that player.
 3. Select one or more of the playback options:
 - **Allow playback in floating window with no title bars:** Opens the media in a separate window without a title bar.
 - **Allow document to set title text in a floating-playback window:** Opens the media in a separate window with a title bar.
 - **Allow playback in full-screen window:** Opens the media in full-screen mode.
- Note:** Membership on the trusted document list is permanent until the list is manually cleared. Choose **Clear** to remove all documents from that list.
4. Choose **OK**.

9.2.2 Controlling Multimedia in Certified Documents

Note: Multimedia and other dynamic content poses a security risk because it could potentially change the document's appearance or allow security holes in multimedia players to adversely impact your system. Participants in certification workflows should consider the source of the document and the security of the workflow before enabling dynamic content.

Whether dynamic content executes in certified documents based on the Trusted Document or Other Document settings depends on two items under your control:

- You can configure a certified document to use the trusted document settings on a per-certificate basis or by using trust anchors. If a signer's certificate chains up to another certificate (a trust anchor) that allows multimedia, then multimedia will run in that certified document. For example, some enterprises may issue a MyCompany certificate that allows dynamic content. If all employee certificates use MyCompany as a trust anchor, then they can send and receive certified documents within the company that could contain working multimedia.
- If the certificate trust settings allow dynamic content, the Multimedia Trust Manager's **Trusted documents** settings are used.
- If the certificate trust settings do not allow dynamic content, the Trust Manager's **Other Documents** settings are used, UNLESS the document has already been added to the trusted documents list.
- You can configure a certified document to always use the trusted document settings regardless of certificate trust levels by adding it to the Trusted Documents list.

Preventing Multimedia Playback in Certified Documents

To prevent dynamic content from playing in any certified document do one of the following:

- Never allow multimedia: Uncheck **Allow multimedia operations** in the Trust Options panel for both trusted and untrusted documents as described in ["Configuring Multimedia Trust Preferences" on page 137](#).

- Never allow multimedia for untrusted documents: Never trust any certificate for dynamic content and clear your trusted document list. Then configure your Other Document multimedia settings to **Never** or **Prompt**.

Note: There is no way to guarantee that multimedia won't play based on the trusted document list and certificate trust level alone. Application preferences always override these restrictions.

9.3 Setting JavaScript Options

9.3.1 High Privilege JavaScript Defined

High privilege JavaScripts are Acrobat methods with security restrictions. These are marked by an "S" in the third column of the quick bar in the *JavaScript for Acrobat API Reference*. These methods can be executed only in a privileged context, which includes the console, batch, menu, and application initialization events. All other events (for example, page open and mouse-up events) are considered non-privileged.

The description of each security-restricted method indicates the events during which the method can be executed. Beginning with Acrobat 6.0, security-restricted methods can execute in a non-privileged context if the document is certified and the certifier's certificate is trusted for executing embedded high privilege JavaScript.

In Acrobat versions earlier than 7.0, menu events were considered privileged contexts. Beginning with Acrobat 7.0, execution of JavaScript through a menu event is no longer privileged. You can execute security-restricted methods through menu events in one of the following ways:

- By going to **Edit > Preferences > JavaScript** and checking the item named **Enable menu items JavaScript execution privileges**.
- By executing a specific method through a trusted function (introduced in Acrobat 7.0). Trusted functions allow privileged code—code that normally requires a privileged context to execute—to execute in a non-privileged context. For details and examples, see `app.trustedFunction` in the *JavaScript for Acrobat API Reference*.

9.3.2 Javascript and Certified Documents

Whether JavaScript runs in certified documents depends on whether you have explicitly trusted the certifier's digital ID certificate (directly or indirectly by trusting an issuer on the certificate chain) for that action. You can control script behavior on a per-certificate basis or by using trust anchors. If a signer's certificate chains up to another certificate (a trust anchor) that allows high privileged JavaScript, then high privileged JavaScript will run in that document. For example, some enterprises may issue a MyCompany certificate that allows high privileged JavaScript. If all employee certificates use ExampleCompany as a trust anchor, then they can send and receive certified documents within the company that contain working JavaScript.

If you need to enable JavaScript in certified documents, set certificate trust.

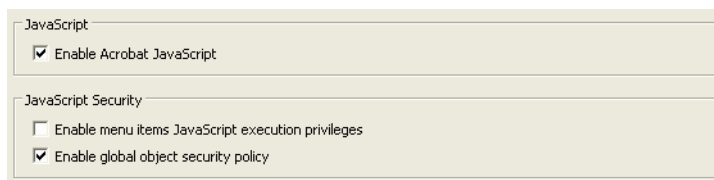
Tip: Because scripts could potentially change the document's appearance or allow attackers access to your system, participants in certified workflows should consider the source of the document and the security of the workflow before enabling this option.

For unsigned documents, you can control JavaScript from the Preferences panel.

To block or allow execution of all JavaScript from the menu bar:

1. Choose one of the following:
 - Acrobat and Adobe Reader (Windows): **Edit > Preferences > JavaScript**
 - Acrobat and Adobe Reader (Macintosh): **(Application) > Preferences > JavaScript**
2. Check or uncheck **Enable menu items JavaScript execution privileges**.
3. Choose **OK**.

Figure 99 JavaScript Security option



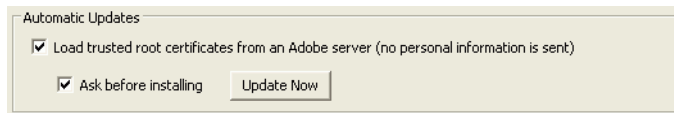
9.4 Adobe Trusted Identity Updates

In order to facilitate workflows that use certificates, Adobe occasionally sends new certificates configured as trust anchors to application users. These certificates allow you to validate signatures that are signed with certificates that chain up to those trusted certificates. In other words, you can validate those signatures without the extra steps of trusting each signer's certificate or manually configuring another trust anchor. These certificates are carefully selected and generally belong to major organizations such as countries or government agencies.

The application default is to check for updates and then ask if you would like to install them. However, you can modify this behavior as follows:

1. Choose one of the following:
 - Acrobat and Adobe Reader (Windows): **Edit > Preferences > Trust Manager**
 - Acrobat and Adobe Reader (Macintosh): **(Application) > Preferences > Trust Manager**
2. Configure the options as needed:
 - Turn the update off or on.
 - Turn **Ask before installing** option off or on.
 - Choose **Update Now** to get the latest certificates.
3. Choose **OK**.

Figure 100 Automatic updates



9.5 Working with Attachments

Before attempting to modify the application's default behavior, you should understand the default behavior. For details, see the following:

- ["Default Behavior: Black and White Lists" on page 141](#)
- ["Adding Files to the Black and White Lists" on page 144](#)
- ["Resetting the Black and White Lists" on page 144](#)
- ["Allowing Attachments to Launch Applications" on page 145](#)

9.5.1 Default Behavior: Black and White Lists

- [Black Lists and White Lists](#)

Black Lists and White Lists

- The Acrobat family of products always allow you to open and save PDF and FDF file attachments. However, **(Table 4)File types on the white list:** These can be attached and may be opened or saved if the file extension is associated with the requisite program.
- **File types on the black list:** These can be attached, but a warning dialog appears stating that they cannot be saved or opened from the application. No actions are available for these files.
- **File types not on any list:** These can be attached without a warning dialog. Trying to open or save them invokes a dialog which allows the user to perform the action just once or to add them to the good type (white) list or bad type (black) list.

Why Attach a File that's on the Black List?

Table 4 Default prohibited file types

Extension	Description
.ade	Access Project Extension (Microsoft)
.adp	Access Project (Microsoft)
.app	Executable Application
.asp	Active Server Page
.bas	BASIC Source Code
.bat	Batch Processing
.bz	Bzip UNIX Compressed file
.bz2	Bzip 2 UNIX Compressed file (replaces BZ)

Table 4 Default prohibited file types

Extension	Description
.cer	Internet Security Certificate file (MIME x-x509-ca-cert)
.chm	Compiled HTML Help
.class	Java Class file
.cmd	DOS CP/M Command file, Command file for Windows NT
.com	Command
.command	Mac OS Command Line executable
.cpl	Windows Control Panel Extension (Microsoft)
.crt	Certificate file
.csh	UNIX csh shell script
.exe	Executable file
.fxp	FoxPro Compiled Source (Microsoft)
.gz	Gzip Compressed Archive
.hex	Macintosh BinHex 2.0 file
.hlp	Windows Help file
.hqx	Macintosh BinHex 4 Compressed Archive
.hta	Hypertext Application
.inf	Information or Setup file
.ini	Initialization/Configuration file
.ins	IIS Internet Communications Settings (Microsoft)
.isp	IIS Internet Service Provider Settings (Microsoft)
.its	Internet Document Set, International Translation
.jar	Java Archive
.job	Windows Task Scheduler Task Object
.js	JavaScript Source Code
.jse	JScript Encoded Script file
.ksh	UNIX ksh shell script
.lnk	Windows Shortcut file
.lzh	Compressed archive (LH ARC)
.mad	Access Module Shortcut (Microsoft)
.maf	Access (Microsoft)
.mag	Access Diagram Shortcut (Microsoft)
.mam	Access Macro Shortcut (Microsoft)
.maq	Access Query Shortcut (Microsoft)
.mar	Access Report Shortcut (Microsoft)
.mas	Access Stored Procedures (Microsoft)
.mat	Access Table Shortcut (Microsoft)

Table 4 Default prohibited file types

Extension	Description
.mau	Media Attachment Unit
.mav	Access View Shortcut (Microsoft)
.maw	Access Data Access Page (Microsoft)
.mda	Access Add-in (Microsoft), MDA Access 2 Workgroup (Microsoft)
.mde	Access MDE Database file (Microsoft)
.mdt	Access Add-in Data (Microsoft)
.mdw	Access Workgroup Information (Microsoft)
.mdz	Access Wizard Template (Microsoft)
.msc	Microsoft Management Console Snap-in Control file (Microsoft)
.msi	Windows Installer file (Microsoft)
.msp	Windows Installer Patch
.mst	Windows SDK Setup Transform Script
.ocx	Microsoft Object Linking and Embedding (OLE) Control Extension
.ops	Office Profile Settings file
.pcd	Visual Test (Microsoft)
.pkg	Mac OS X Installer Package
.pif	Windows Program Information file (Microsoft)
.prf	Windows System file
.prg	Program file
.pst	MS Exchange Address Book file, Outlook Personal Folder file (Microsoft)
.rar	WinRAR Compressed Archive
.reg	Registration Information/Key for Windows 95/98, Registry Data file
.scf	Windows Explorer Command
.scr	Windows Screen Saver
.sct	Windows Script Component, Foxpro Screen (Microsoft)
.sea	Self-expanding archive (used by Stuffit for Mac files and possibly by others)
.shb	Windows Shortcut into a Document
.shs	Shell Scrap Object file
.sit	Compressed archive of Mac files (Stuffit)
.tar	Tape Archive file
.tgz	UNIX Tar file Gzipped
.tmp	Temporary file or Folder
.url	Internet Location
.vb	VBScript file or Any VisualBasic Source
.vbe	VBScript Encoded Script file
.vbs	VBScript Script file, Visual Basic for Applications Script

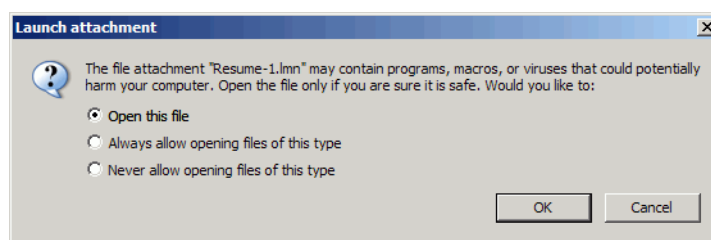
Table 4 Default prohibited file types

Extension	Description
.vsmacros	Visual Studio .NET Binary-based Macro Project (Microsoft)
.vss	Visio Stencil (Microsoft)
.vst	Visio Template (Microsoft)
.vsw	Visio Workspace file (Microsoft)
.webloc	Mac OS Finder Internet Location
.ws	Windows Script file
.wsc	Windows Script Component
.wsf	Windows Script file
.wsh	Windows Script Host Settings file
.zip	Compressed Archive file
.zlo	ZoneLabs ZoneAlarm Mailsafe Renamed .PIF file
.zoo	An early compressed file format

9.5.2 Adding Files to the Black and White Lists

4. [Table 4](#) can be extended one at a time as each attached file is opened. Administrators can modify the registry directly (refer to the *Acrobat Security Administration Guide*). When the Launch Attachment dialog appears, choose one of the following ([Figure 101](#)):
- **Open this file:** Opens the files without changing the registry list.
 - **Always allow opening files of this type:** Adds the
 - **Never allow opening files of this type:** Adds the file type to the black list and does not open it.

Figure 101 Launch Attachment dialog



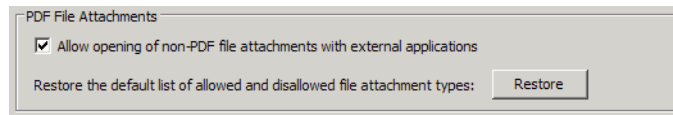
9.5.3 Resetting the Black and White Lists

Because the registry list could grow over time and users do not have direct access to the lists through the user interface, resetting the list to its original state may result in the highest level of security.

To reset the black and white lists:

1. Choose **Edit > Preferences** (Windows) or **Acrobat > Preferences** (Macintosh).
2. Select Trust Manager in the Categories panel.
3. Choose **Restore** ([Figure 102](#)).

Figure 102 Attachment panel in Trust Manager



9.5.4 Allowing Attachments to Launch Applications

The Trust Manager enables users to control whether or not non-PDF attachments can open with other applications. By default, this option is enabled so that common file types such as .doc (not on the application's black list) can be easily opened in the appropriate application.

To set attachment preferences:

1. Choose **Edit > Preferences** (Windows) or **Acrobat > Preferences** (Macintosh).
2. Select Trust Manager in the Categories panel.
3. Configure **Allow opening of non-PDF file attachments with external applications** (Figure 102):
 - **Checked:** Default. The application uses its stored black list to determine whether Acrobat should let the attachment invoke the launch an application action so the attachment can be opened.
 - **Unchecked:** Clicking or opening an attachment will never result in launching it's associated viewing application. Use this option if a higher level of security is needed.

9.6 Controlling Access to Referenced Files and XObjects

Your application can inform you when a PDF file is attempting to access external content identified as a stream object by flags as specified in the *PDF Reference*. For example, a URL might point to an image external to the document. Only PDF developers create PDF files with streams, so you may not need to enable access to external content.

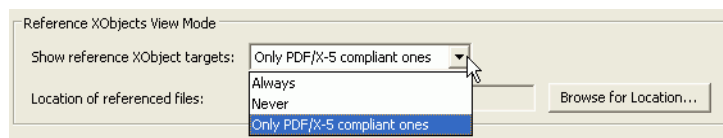
Silently transmitting data represents a security risk since malicious content can be transferred whenever the application communicates with an external source. Therefore, you may want to disable this feature.

To configure external content access:

1. Choose **Edit > Preferences** (Windows) or **Acrobat > Preferences** (Macintosh).
2. Select Page Display in the Categories panel.
3. Configure the Reference XObjects View Mode panel:
 - Set **Show reference XObject targets** to:
 - Always
 - Never
 - Only PDF/X-5 compliant ones
4. Set the location of referenced files (if any).

5. Choose **OK**.

Figure 103 Resource access



9.7 Internet URL Access

Your application can inform you when a PDF file is attempting to connect to an Internet site. Opening a Web page represents a security risk because malicious content can be transferred whenever the application communicates with the Internet. In addition to visible links in a PDF document, form fields can contain hidden JavaScript calls that open a page in a browser or silently requests data from the Internet.

Tip: This feature interacts with the new Security (Enhanced) preference feature. URLs that are set as privileged locations are exempt from enhanced security restrictions even if enhanced security is on.

You can control Internet access via the Manage Internet Access dialog ([Figure 105](#)). Controls are provided for the following:

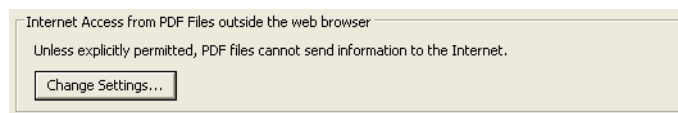
- [“Turning Internet Access Off and On” on page 146](#)
- [“Allowing and Blocking Specific Web Sites” on page 147](#)

9.7.1 Turning Internet Access Off and On

To block or allow all Web sites:

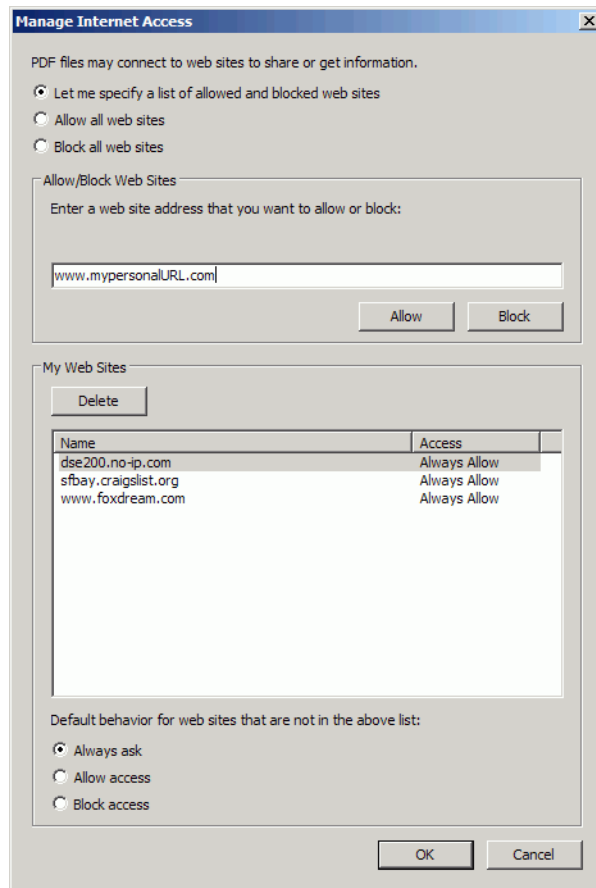
1. Choose **Edit > Preferences** (Windows) or **Acrobat > Preferences** (Macintosh).
2. Select Trust Manager in the Categories panel.
3. Choose **Change Settings** in the **Internet Access...** panel.

Figure 104 Internet access panel



4. Choose **Allow all web sites** or **Block all web sites** ([Figure 105](#)).
5. Choose **OK**.

Figure 105 Manage Internet Access dialog



9.7.2 Allowing and Blocking Specific Web Sites

The Acrobat family of products maintain a white and black list of URLs called the *Trust List*. Users can specify whether or not URL access is allowed on a global or per-URL basis. For URLs that aren't explicitly trusted or blocked (they are not on the white or black list), a warning appears whenever a document tries to access the Internet (Figure 107). When you check **Remember my action for this site**, the site is added to your URL white or black list.

Figure 106 Blocked URL alert

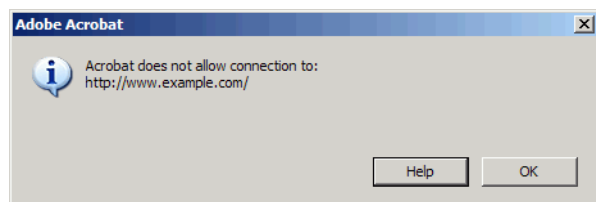
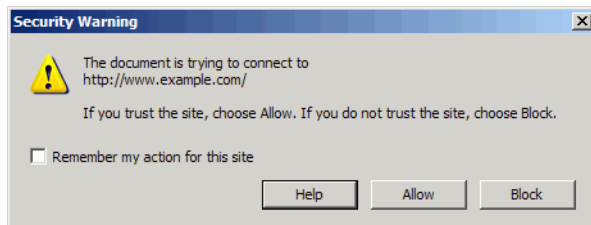


Figure 107 External connection warning



To configure Internet resource access on a per-URL basis, add specific Web sites to the black and white lists:

1. Choose **Edit > Preferences** (Windows) or **Acrobat (or Adobe Reader) > Preferences** (Macintosh).
2. Select Trust Manager in the Categories panel.
3. Choose **Change Settings** in the **Internet Access...** panel.
4. Choose **Let me specify a list of allowed and blocked web sites**.
5. Configure the black and white lists:
 - Add a URL to the URL fields and choose **Allow** or **Block**.
 - Choose a URL already in My Web Sites panel and choose **Delete**.
6. Select an option from the **Default behavior for web sites that are not in the above list**:
 - **Always ask**: You will be prompted to allow or block access for URLs not in the Trust List.
 - **Allow access**: URLs not in the Trust List will always be accessible.
 - **Block access**: URLs not in the Trust List will never be accessible.
7. Choose **OK**.

Note: This document was formerly an FDF user guide.

Security settings can be complex, and more importantly, they are often critical components of digital signature and document security workflows. For this reason, it's often necessary to migrate and even share settings across one or more machines. There are two methods available:

- **Security Setting Import and Export:** One of Acrobat 9.0's major new security features includes the ability to import and export security settings via .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:
 - All 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.
 - Security. Files can be signed and encrypted.
 - Files can be used to backup and restore settings, to distribute settings in a workgroup or enterprise, and to send specific information to another user.
- **Sharing Settings & Certificates with FDF:** FDF files are useful for importing and exporting a specific type of setting such as trust anchors, timestamps, directory servers, and so on.

10.1 Security Setting Import and Export

Acrobat 9.0 introduces a new feature that helps users and organizations migrate existing security settings through version upgrades and across multiple machines. Unlike FDF files, the new .acrobatsecurity settings file supports the import and export of all settings including digital ID data, trust, server details, signing preferences, and so on. Settings can only be exported from Acrobat but settings can be imported by both Acrobat and Adobe Reader.

10.1.1 Exporting Security Settings to a File

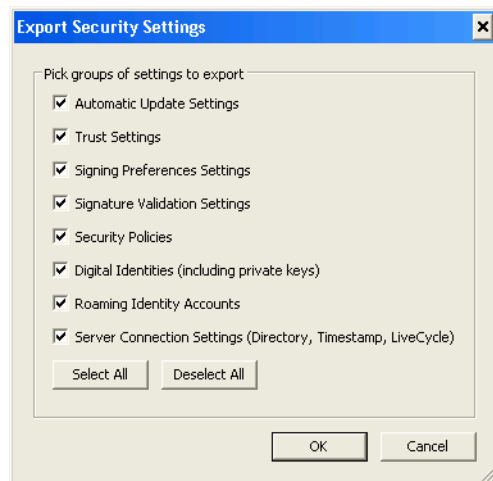
Settings can only be exported from Acrobat.

1. Choose **Advanced > Security > Export Security Settings**.
2. Check or uncheck the settings you would like to export.

Note: Whether you export or import settings via an FDF file or an .acrobatsecurity setting file, the actual settings are the same. Details about each individual setting are found in the FDF section as well as elsewhere in this document.

Choose **OK**.

Figure 108 Security settings: Export dialog



3. When the detailed Export Security Settings dialog appears, review the settings again.
4. If you would like to include or exclude any settings, highlight the setting and choose the **Include/Exclude Setting** button.
5. Choose **Export**.
6. Choose an encryption method. Encrypting the file ensures that the settings can't be viewed by anyone other than the intended recipients.

Figure 109 Security settings: Encryption method



7. Follow the dialog instructions which will vary with your choice of the document security method (password security or certificate security).
8. Choose **OK**.
9. You will be required to certify the file by signing it with a certification signature. When the certification workflow begins, choose **OK**.
10. Sign and save the file.

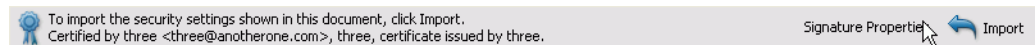
10.1.2 Importing Security Settings from a File

Settings can be imported by both Acrobat and Adobe Reader.

To import security settings:

1. Choose **Advanced > Security > Import Security Settings**.
2. Browse to an .acrobatsecuritysettings file.
3. Choose **Open**.
4. .acrobatsecuritysettings files must be certified and are therefore signed. You can verify the signer's identity by choosing the **Signature Properties** in the Document Message Bar and reviewing the signer's details.

Figure 110 Security settings: Document message bar

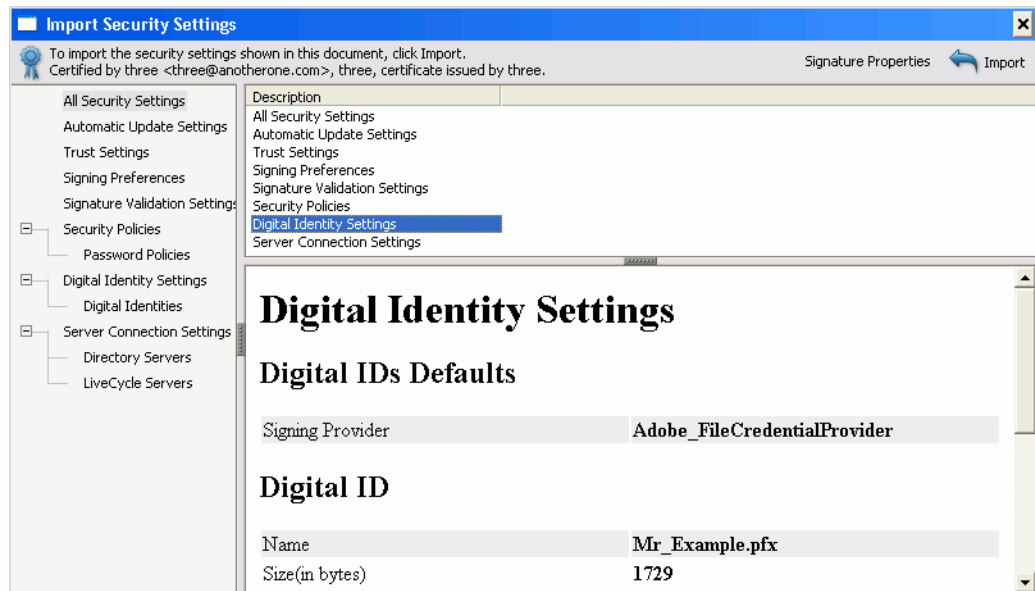


5. Review the settings carefully.

Note: Whether you export or import settings via an FDF file or an .acrobatsecurity setting file, the actual settings are the same. Details about each individual setting are found in the FDF section as well as elsewhere in this document.

Caution: The settings in the imported file will overwrite your current settings. Be sure to verify you're getting the correct settings and that they are coming from a trusted source.

Figure 111 Security settings: Import from a file panel



6. Choose **Import**.
7. If the settings you imported included Digital IDs, you must log into each such ID to complete its installation. If there were Digital IDs then a dialog appears asking if you'd like to open the Security Settings Console and log in to the digital IDs you just imported. Choose **Yes** or **No**.

Note: For security reasons, .acrobatsecuritysettings files do not carry the digital ID passwords. Before you can use any of the digital IDs you just imported, you must log in to each ID. You can do it now or later.

Figure 112 Security setting import: Success dialog

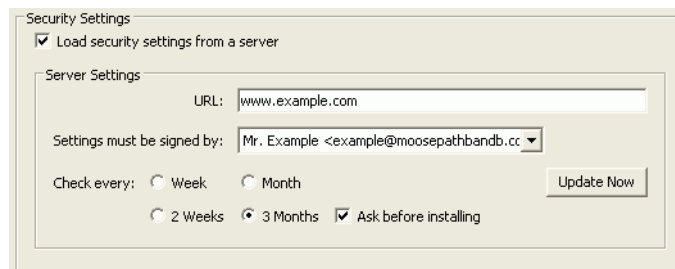


10.1.3 Importing Security Settings from a Server

If your organization distributes security settings periodically, you can set up Acrobat to regularly check for updates to these policies. Server-based security is set up by an administrator who provides the URL from which to get security updates. Once the application is configured, Acrobat will periodically poll the server (the default time is every three months) via http or https.

1. Choose **Edit > Preferences > Security**.
2. Check **Load security settings from a server**.
3. Enter the server address in the URL field.
4. Select a signing certificate if any. The .acrobatsecurity file will be signed with a certified signature. In order to install the file, you will need validate the signature.
5. Specify how often you want to check for security updates.
6. Select **Ask Before Installing** to be notified prior to installing new settings.
7. When the acrobatsecurity file opens, follow the instructions as described in [“Importing Security Settings from a File” on page 150](#).

Figure 113 Security setting preferences for server import



10.2 Sharing Settings & Certificates with FDF

Note: The first time you receive settings, you may not have the certificate of the signer of the settings file. This will result in some additional dialogs asking if you are sure you trust the source of the

settings file. Once installed, the settings file should include the proper certificate so these additional questions will be avoided in subsequent updates.

Acrobat and Adobe Reader support the use of FDF files to exchange data between the Acrobat family of client and server products. FDF files use a .fdf extension, and like .pdf, it is registered by Adobe so that the required application is used to open these files via a browser or file explorer. Acrobat provides the following FDF features:

- Import and export of digital ID certificates.
- Import and export of server settings for an Adobe LiveCycle Rights Management Server, LDAP directory servers, roaming credential servers, and timestamp servers.

Whether the file is located on a network or emailed, FDF file recipients simply double click on a FDF file to import its data automatically via the FDF import wizard, thereby eliminating the need for error prone, manual configuration.

FDF files provide individuals and businesses with many opportunities for streamlining workflows. For example:

- Alice wants to email her certificate to Bob and wants Bob to reply with his certificate. Alice chooses **Request Contact** in the Trusted Identity Manager. The workflow generates and emails an FDF file that can contain her certificate, a request for Bob's certificate, and Alice's return email address.
- Alice needs to encrypt documents for a number of people in her organization. An administrator sends her an FDF file that contains a large group of contacts. When Alice opens the FDF file, she is walked through the FDF Data Exchange UI wizard so that she can import these contacts into her Trusted Identities list.
- A server wants a copy of Bob's certificate so that the server can encrypt documents for Bob. The server generates an FDF file that contains a certificate request and a return URL address. When Bob downloads the FDF file from the server, he is walked through the FDF Data Exchange UI wizard where he can respond by allowing his certificate to be returned.
- A company needs to distribute its trusted certificate to customers so that they can verify that the company's documents are authentic. A server or administrator creates an FDF file that contains the trusted certificate and posts it on a Web server that hosts a Web page with a link to the file. When customers download the file, they are asked whether they wish to add this certificate to the Trusted Identity list and are given the ability to set the certificate's trust level.

For more information, refer to the following:

- [FDF Files and Security](#)
- [Importing Application Settings with FDF Files](#)
 - ["Responding to an Email Request for a Digital ID" on page 164](#)
 - ["Importing Someone's Certificate" on page 166](#)
 - ["Importing Multiple Certificates" on page 167](#)
 - ["Importing Timestamp Server Settings" on page 169](#)
 - ["Importing Directory Server Settings" on page 171](#)
 - ["Importing Adobe LiveCycle Rights Management Server Settings" on page 172](#)
 - ["Importing Roaming ID Account Settings" on page 173](#)
 - ["Importing a Trust Anchor and Setting Trust" on page 175](#)
- [Exporting Application Settings with FDF Files](#)

- [“Distributing a Trust Anchor or Trust Root” on page 155](#)
- [“Setting the Certificate Trust Level” on page 158](#)
- [“Exporting Your Certificate” on page 158](#)
- [“Emailing Your Certificate” on page 159](#)
- [“Saving Your Digital ID Certificate to a File” on page 160](#)
- [“Requesting a Certificate via Email” on page 161](#)
- [“Emailing Server Details” on page 162](#)
- [“Exporting Server Details” on page 163](#)

10.2.1 FDF Files and Security

FDF files are data exchange files. Like acrobatsecurity files, they help you move certificate, server, and other data from one machine to another. This data transfer usually involves some mechanism such as data injection into a PDF form field, installing files, executing a script, and so on. These actions represent a potential security risk, and in some environments that risk may be unacceptable. Acrobat therefore provides a new security feature that, when turned on, disables some FDF functionality unless those FDF files originate from a specifically privileged file, folder, or server.

The new feature is called Enhanced Security and may be enabled or disabled by choosing **Edit > Preferences > Security (Enhanced)**. [Table 5](#) lists the high level rules defining FDF behavior.

Tip: If you need to configure your environment for enhanced security or need to troubleshoot FDF workflows that may not be working as expected, see [“Enhanced Security” on page 132](#).

Table 5 Rules for opening a PDF via FDF

Action	FDF location	PDF location	8.x behavior	9.x behavior
Opening a target PDF	local	local	PDF opens and no authentication required.	Same.
Opening a target PDF	local	http server	PDF opens	User authorization required unless trusted via enhanced security feature.
Opening a target PDF	https server	http server	PDF opens and no authentication required.	Same.
Opening a target PDF	https server	local	Blocked	Http hosted FDFs cannot open local files.
Data injection	n/a	n/a	Allowed	Allowed if: <ul style="list-style-type: none"> • Data returned via a form submit with url#FDF. • FDF has no /FDF key. • cross-domain policy permits it.

Table 5 Rules for opening a PDF via FDF

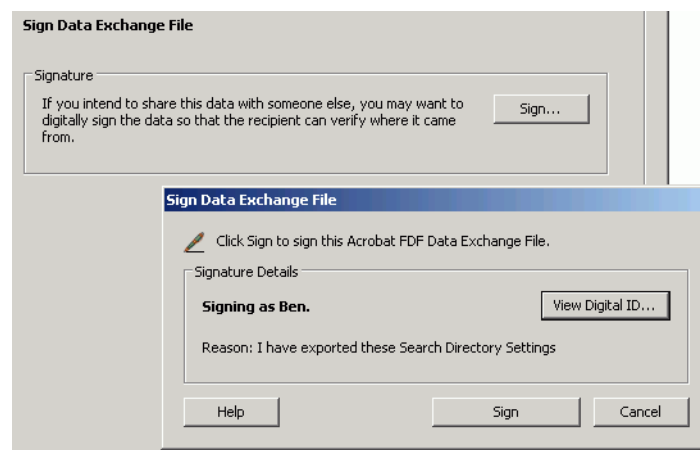
Action	FDF location	PDF location	8.x behavior	9.x behavior
Data injection	server	browser	Allowed	Allowed if: <ul style="list-style-type: none"> Link to PDF contains #FDF=url. FDF has no /FDF key. x-domain policy permits it.
Data injection	server	Application	Allowed	Allowed if: <ul style="list-style-type: none"> PDF makes EFS POST/GET and FDF sends data in https response to same PDF. x-domain policy permits it.
Data injection	Varied	Varied	Allowed	Authorization required if enhanced security is on and document is not set as a privileged location.
Script injection	Any	Any	Allowed	Injection is blocked unless if enhanced security is on and FDF is not in a privileged location.

10.2.2 Exporting Application Settings with FDF Files

FDF files can be created by administrators, end users, and even a server. It is a good idea to sign FDF files so that recipients of the file can establish a level of trust for the contents of the FDF file. For example, when an FDF file is signed, the **Accept the level of trust specified by the signer for all contacts in this file** checkbox becomes enabled, thereby allowing the importer to accept the level of trust you have specified.

Note: Recipients won't be able to validate your signature unless you have previously sent them your digital ID certificate or your certificate was issued by someone they already trust.

Figure 114 Signing an FDF file



10.2.2.1 Distributing a Trust Anchor or Trust Root

Distributing a trusted certificate from Acrobat involves wrapping one or more certificates in an FDF file and making it available to other users via email, a network directory, or a Web site. Recipients simply click on the file or a link to the file to open the Acrobat wizard which downloads and/or installs the certificate.

Certificate Chains and Trust Anchors /Roots

Certificates usually exist as part of a hierarchy or “chain” of certificates, and part or all of the chain can be wrapped in an FDF file. The bottom-most and end user certificate (yours) is called an “end entity” (EE) certificate. The top-most certificate, (the root) is typically belongs to a trusted Certificate Authority (CA). 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. Acrobat enables users to specify one or more of the certificates in a chain as trusted for specific operations. Thus, an EE certificate could have one or more trust anchors (trusted ICAs) that chain up to a the top-most CA certificate which is the primary trust anchor or “trusted root.”

A typical chain might include your certificate, your company’s ICA, and a root CA. Certificates inherit trust from certificates on the root end of the chain. For example, if the root certificate is trusted, then any certificates chaining to the that root will also be trusted. Some organizations have their own root CA or use an ICA certificate that is issued by an external CA and make these the trust anchors for their employees.

It is a common practice to trust certificates as high up in the chain as is reasonable since revocation checking starts at the chain bottom and continues until it reaches a trust anchor. Revocation checking occurs until reaching a certificate that is absolutely trusted by you or your organization. It also allows users to trust other certificates that chain up to the same root. The trust anchor is often an ICA for example, since if the root is issued by a company such as VeriSign, it might not be wise to make it a trust anchor as that tells Acrobat to trust the millions of certificates that chain up to VeriSign.

Distributing and installing ICA or CA trust anchors to a user or group of users allows them to:

- Distribute certified or signed documents to partners and customers.
- Help document recipients validate the signatures of document authors.

Exporting a Trust Anchor

When Acrobat exports a certificate, it automatically exports other selected certificates in that certificate’s chain and includes them in the FDF file.

1. Choose **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Manage Trusted Identities**.
2. Choose **Certificates** in the **Display** drop-down list.

In addition to this method, you can also display the certificate from any signature or certificate security method workflow where a **Show Certificate** or **Certificate Details** button appears, such as the Signature Properties dialog.

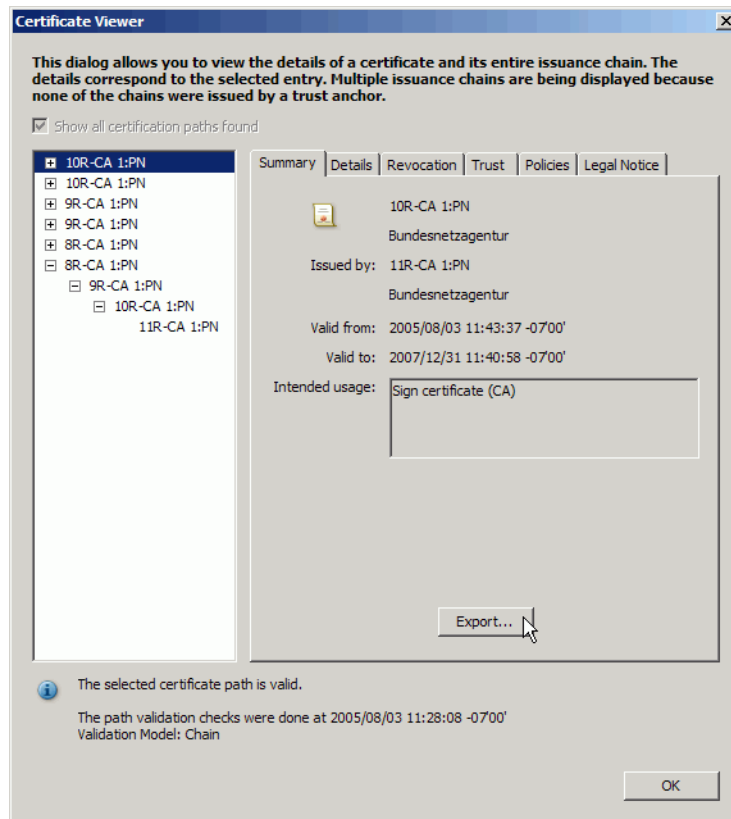
3. Select the certificate ([Figure 116](#)).

Note: In the unlikely event that you can sign the FDF file with a signature the recipient can validate (they will use a different certificate than the one you are exporting), set the certificate’s trust level before exporting it. For details, see [“Setting the Certificate Trust Level” on page 158](#)

Tip: You could just choose **Export** and bypass the following two steps. However, exporting the certificate from the Certificate Viewer allows you to see the entire certificate chain where you can select all or part of it.

4. Choose **Show Certificate**. The Certificate Viewer displays the certificate.
5. Select a certificate in the chain that appears in the left-hand window.

Figure 115 Selecting a certificate chain for export



6. Choose **Export**.
7. Choose one of the following:
 - **Email the data to someone:** Emailing the data automatically creates an FDF file that other Adobe product users can easily import.
 - **Save the exported data to a file:** Acrobat FDF Data Exchange. FDF is a format recognized by the Acrobat family of products.
8. Choose **Next**.
9. (Optional) If the Identity Information dialog appears, enter the your email address and any other information. If you have already configured your identity details, this screen may not appear. For details, see [“Setting Identity Information” on page 14](#).
10. **Do not sign** if the certificate you use to sign uses the same trust anchor or you are distributing. Since recipients do not have this certificate yet, they will not be able to validate your signature.

Note: Signing the FDF will only be useful if you have a digital ID that the recipient has already trusted (uses a trust anchor OTHER than the one you are currently distributing). The FDF file recipients must also already have that digital IDs certificate so that they can validate your signature without relying on the certificate you are currently sending. This workflow is uncommon, but it does allow recipients to automatically inherit your predefined trust settings for the certificate embedded in the file.
11. Choose **Next**.

- Continue with the workflow until the trusted root is emailed or placed in a directory where your intended recipients can find it.

Providing Instructions to the Trusted Root Recipients

For details, see [“Importing a Trust Anchor and Setting Trust” on page 175.](#)

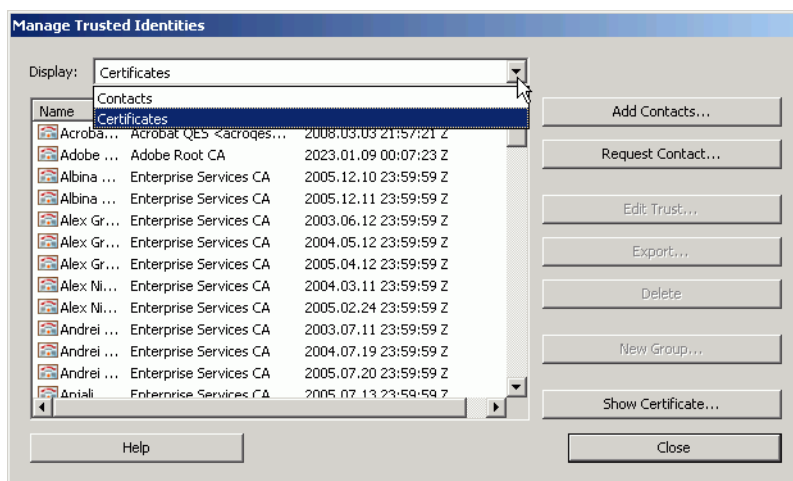
10.2.2.2 Setting the Certificate Trust Level

Note: This section is only relevant for trust anchor’s in FDF files that are signed with a trusted signature. This is an unlikely scenario, since the trust anchor distributor is probably using the same trust anchor that is being distributed and the recipient doesn’t have it yet. Most users will likely need to manually set the imported certificate’s trust level.

When distributing a trusted root in a signed file that the FDF recipient can validate, set the certificate trust level:

- Choose **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Manage Trusted Identities**.
- Choose **Certificates** in the **Display** drop-down list.

Figure 116 Certificates in the Trusted Identities list



- Highlight the needed certificate.
- Choose **Edit Trust**.
- Display the Trust tab.
- Set the trust level as described in [“Importing a Trust Anchor and Setting Trust” on page 175.](#)

10.2.2.3 Exporting Your Certificate

You can use FDF files to export your certificate so that others can import it into their list of trusted identities. This enables them to encrypt documents for you and validate your signature for documents that you digitally sign.

- Before users receiving your signed document can validate your signature, they must receive the your certificate or one above it in the trust chain.

- Before users can encrypt a document for you with certificate encryption, they must have access your certificate.

Certificates can be emailed or saved to a file for later use. There are two ways to export a certificate:

- To export a certificate from the list in the Security Settings Console, refer the following:
 - [“Emailing Your Certificate” on page 159](#)
 - [“Saving Your Digital ID Certificate to a File” on page 160](#)
- To export any certificate displayed in the Certificate Viewer, choose **Export** on the Summary tab.

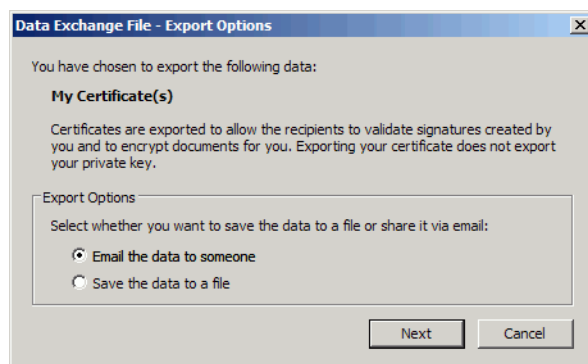
10.2.2.4 Emailing Your Certificate

If you do not have an email program on your machine, save the data to a file as described in [“Saving Your Digital ID Certificate to a File” on page 160](#) and then send the file as an attachment using your web-based email program.

To email a digital ID certificate:

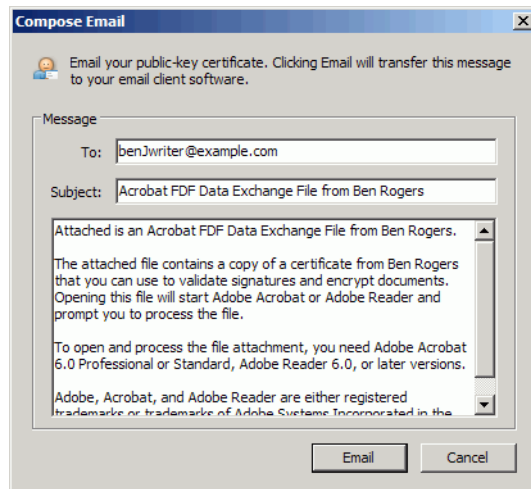
1. Choose **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Security Settings**.
2. Select **Digital IDs** in the left-hand tree.
3. Highlight an ID in the list on the right. If you have more than one, choose the one that is appropriate for the usage context. For example, send your company-issued ID to those you do business with.
4. Choose **Export**.
5. Choose **Email the data to someone** ([Figure 117](#)).

Figure 117 Digital ID: ID export options



6. Choose **Next**.
7. Enter the recipient's email address and any other optional information.

Figure 118 Emailing your certificate



8. Choose **Email**.
9. When the email program opens, send the email.

Note: Some email problems only queue messages to be sent. You may need to start your email client program to cause the message to actually send.

10.2.2.5 Saving Your Digital ID Certificate to a File

To save a digital ID certificate to a file:

1. Choose **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Security Settings**.
2. Select **Digital IDs** in the left-hand tree.
3. Highlight an ID in the list on the right.
4. Choose **Export**.
5. Choose **Save the exported data to a file** (Figure 117).
6. Choose a file type:
 - **Acrobat FDF Data Exchange:** FDF files enable the easy exchange of data between any Acrobat family of products.
 - **Certificate Message Syntax - PKCS#7:** Save the file as a PKCS7 file. Use this format when the data will be imported into a non-Adobe store such as the Macintosh key store or Windows Certificate Store.
7. Choose **Next**.
8. Browse to a file location and choose **Save**.
9. Choose **Next**.
10. Review the data to export and choose **Finish**.

10.2.2.6 Requesting a Certificate via Email

When you request digital ID information from someone, the application automatically attaches to the email an FDF file containing your contact information and certificate.

To request a certificate from someone:

1. Choose **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Manage Trusted Identities**.
2. Choose **Request Contact**.

Figure 119 Emailing a certificate request

3. Confirm or enter your identity so that the recipient can identify you. The identity panel is prepopulated if the information has been previously as described in [“Setting Identity Information” on page 14](#).
4. Choose **Include My Certificates** to allow other users to add your certificate to their list of trusted identities.
5. Choose whether to email the request or save it as a file.
6. Choose **Next**.
7. Select one or more digital IDs to export. Highlight contiguous IDs by holding down the Shift key. Highlight non-contiguous IDs by holding down the Control key.

Figure 120 Certificates: Selecting a digital ID for export

Name	Issuer	Expires
Ben	Ben	2010.08.10 23:08:02 Z
Fred Smith	Fred Smith	2004.12.08 11:20:18 Z
Joe Smith	Joe Smith	2005.11.12 18:36:23 Z
Johnny Rotten	Johnny Rotten	2005.02.08 20:36:56 Z
Rose ValidTestCA	CDS QE CA	2009.01.26 08:00:00 Z

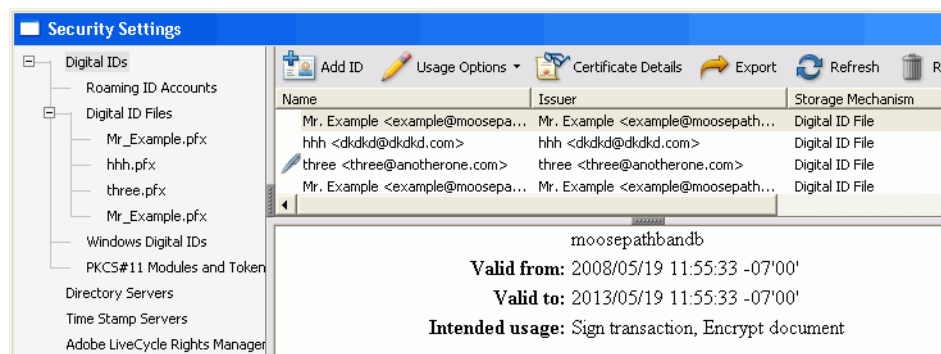
8. Choose **Select**.
9. The next step varies depending on whether you chose to email the ID:
 - **If you chose Email:** Enter the person's email address in the Compose Email dialog and choose **Email**. Send the email message when it appears in the launched email application with the certificate request attached.
 - **If you chose Save as file:** Choose a location for the certificate file Export Data As dialog. Choose **Save**, and then choose **OK**. Tell the intended recipient(s) where to find the file.

10.2.2.7 Emailing Server Details

Adobe LiveCycle Rights Management Server, directory server, roaming credential server, and timestamp server details can be exported to an FDF file for distribution to one or more people. Server information sent via an email resides in an attached FDF file. To send directory server details in an email:

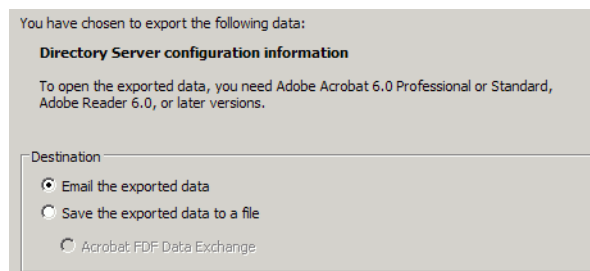
1. Choose **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Security Settings**.
2. Select a server category from the left-hand list.
3. Select a server from the right-hand panel.
4. Choose **Export**.

Figure 121 Security Settings menu items



5. Choose **Email the exported data** to email the FDF file.

Figure 122 Digital ID Directory servers: Export destination



6. Choose **Next**.

The Identity panel (Figure 123) will not appear if the information has been previously configured. For details, see “Setting Identity Information” on page 14.

Figure 123 Digital ID Directory servers: Sender’s identify

Your identity information is used with comments, reviews, and digital signatures. Information entered here is secure and not transmitted beyond this application without your knowledge. To modify this information in the future, simply go to the Identity panel in the preferences.

Identity

Login Name: brogers

Name: Neb Whifflesnit

Title: Senior Technical Writer

Organization Name: Acrobat

Organization Unit: CCC

Email Address: neb.whifflesnit@adobe.com

☒ Do not show again

7. Choose **Sign** and complete the signing workflow (Figure 133). Sign FDF files so that recipients of the file can easily trust the file and its contents.
8. Choose **Next**.
9. Enter the email information.

Figure 124 Digital ID Directory servers: Email details

You can specify the contents of the email message to which you will attach the exported data. This information will be sent to your email program.

Message

To: frontallabotomy@bottleinfrontofme.com

Subject: Acrobat FDF Data Exchange File from Neb Whifflesnit

Attached is an Acrobat FDF Data Exchange File from Neb Whifflesnit.

The attached file contains search directory configuration information that can be used to access search identity directories.

Opening this file will start Adobe Acrobat or Adobe Reader and prompt you to process the file.

To open and process the file attachment, you need Adobe Acrobat 6.0 Professional or Standard, Adobe Reader 6.0, or later versions.

10. Choose **Next**.
11. Review the export details.
12. Choose **Finish**.

10.2.2.8 Exporting Server Details

Adobe LiveCycle Rights Management Server, directory server, roaming ID, and timestamp server details can be exported to an FDF file for distribution to one or more people. Server information can be written to a file and saved to any location.

To save server details to a file:

1. Choose **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Security Settings**.
2. Select a server category from the left-hand list.
Note: For roaming ID server settings, choose an account under **Roaming ID Accounts**.
3. Select a server from the right-hand panel.
4. Choose **Export**.
5. Choose **Save the exported data to a file** to save the data in an FDF file that can be shared (Figure 122).
6. Choose **Next**.
The Identity panel (Figure 123) will not appear if the information has been previously configured. For details, see “Setting Identity Information” on page 14.
7. Choose **Sign** and complete the signing workflow (Figure 133). Sign FDF files so that recipients of the file can easily trust the file and its contents.
8. Choose **Next**.
9. Browse to a location in which to save the file.
10. Choose a file name and choose **Save**.
11. Choose **Next**.
12. Review the export details.
13. Choose **Finish**.

10.2.3 Importing Application Settings with FDF Files

There are several ways to import Acrobat and Adobe Reader data from an FDF file:

- By choosing **File > Open**.
- Double clicking on an FDF file (.fdf)

Tip: The first two options above automatically invoke the simplest workflow.

- For FDF digital ID information, importing it into the Trusted Identity Manager.
- For FDF server settings, importing it with the Security Settings Console.

10.2.3.1 Responding to an Email Request for a Digital ID

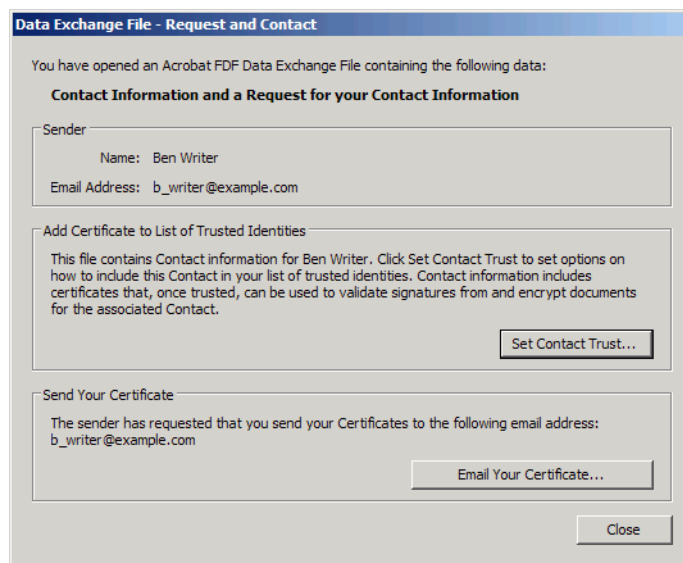
There may be times when someone else needs your digital ID to verify your signature or encrypt a file for you to decrypt (for example, when applying certificate security). To do either, they need access to the public part of your digital ID so that it can be added to their trusted identities list. One way someone can get your ID is to request it in an email.

To request your certificate, a user will simply choose **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Manage Trusted Identities** and then choose **Request Contact**. Acrobat automatically attaches an FDF file with their public certificate to an email that requests your digital ID. The workflow is essentially a digital ID “trade” that allows two users to exchange digital IDs. You must have a digital ID before responding to the request.

To respond to an email digital ID request:

1. Double click the attached FDF file.
2. Choose **Email your Certificate**.

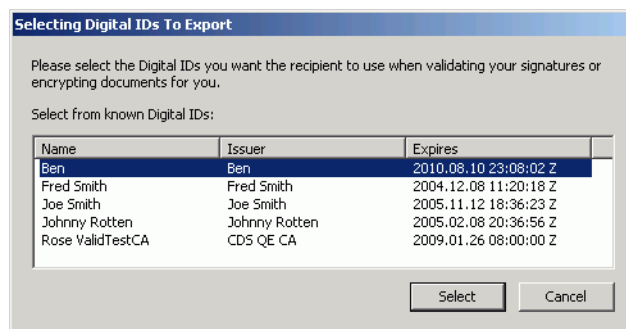
Figure 125 Emailing your certificate



3. Choose a digital ID from the list of existing digital IDs.

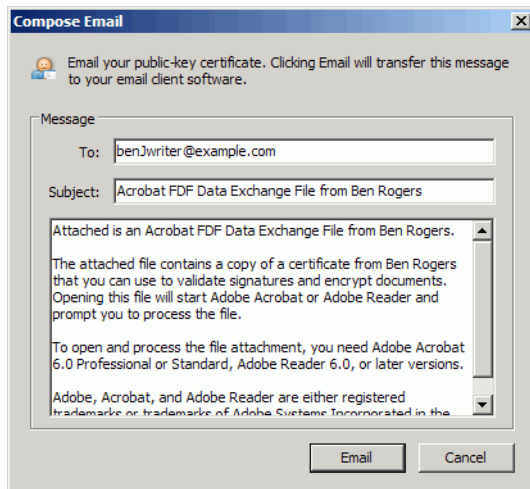
Note: If you do not have a digital ID or choose **Cancel**, an alert appears that says "A certificate was not selected for export." Exit the workflow and get a digital ID.

Figure 126 Selecting a digital ID



4. Choose **Select**.
5. Review the email details. You can edit the To, Subject, and Body fields (Figure 127).
6. Choose **Email**.
7. Send the email through your mail application.

Figure 127 Emailing your certificate



10.2.3.2 Importing Someone's Certificate

You can use an FDF file to import someone's certificate into your list of trusted identities. This enables you to validate their signature and encrypt documents with their public key so only that intended recipient can open it.

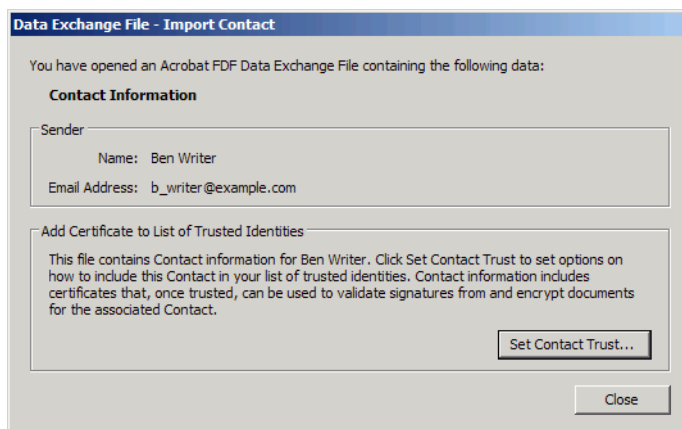
Tip: Importing this information ahead of time enables you to configure your trusted identities list before needing to validate a signature or encrypt a document for someone.

To add someone's certificate to your list of trusted identities:

1. Click on the FDF file or from Acrobat or Adobe Reader choose **File > Open**. The digital ID certificate may be sent directly from Acrobat as an email attachment or may reside in a networked directory.
2. Review the sender's information when the Import Contact dialog appears.

Note: If the file is signed, then the Import Contact dialog will also have a Signature panel as shown in [Figure 129](#).

Figure 128 Certificates: Contact Information



3. Choose **Set Contact Trust**.
4. When the Import Contact Settings dialog appears, configure the Trust and Policy Restrictions. For details, see ["Importing a Trust Anchor and Setting Trust" on page 175](#).
5. Choose **Certificate Details**.
6. Choose the Details tab.
7. In the Certificate data panel, scroll to MD5-digest and SHA-1 digest and note the fingerprint numbers.
8. Contact the certificate's originator and verify the fingerprints are correct.
9. Choose **OK**.
10. Choose **OK**.
11. Choose **Close**.

10.2.3.3 Importing Multiple Certificates

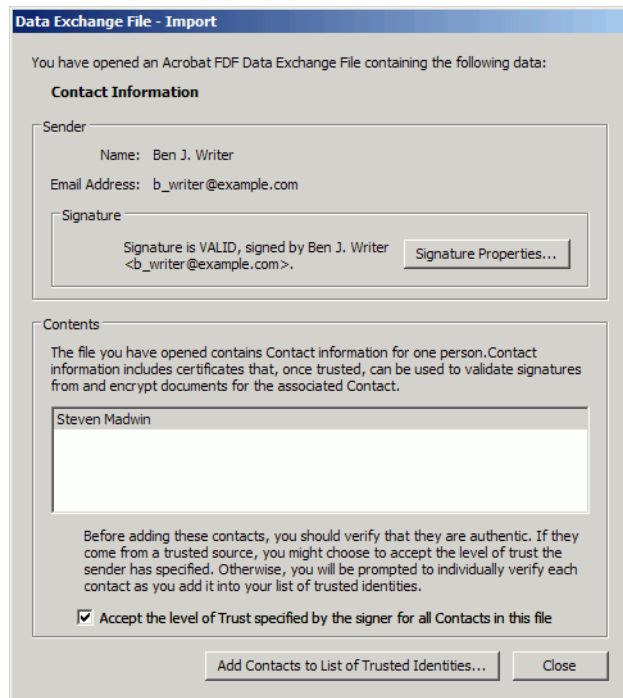
You can use an FDF file to import multiple certificates or a company-wide address book into your list of trusted identities. This enables you to encrypt a document using the public key of the intended recipient so that only they can open it.

Tip: Importing this information ahead of time enables you to configure your trusted identities list before needing to validate signature or encrypt a document to those identities. Administrators can create a company-wide address book and can export it to an FDF file for distribution throughout a company via a network or email.

To add multiple certificate to the trusted identities list all at once:

1. Click on the FDF file or from Acrobat or Adobe Reader choose **File > Open**. The digital ID certificate may be sent directly from Acrobat as an email attachment or may reside in a networked directory.

Figure 129 Importing multiple certificates

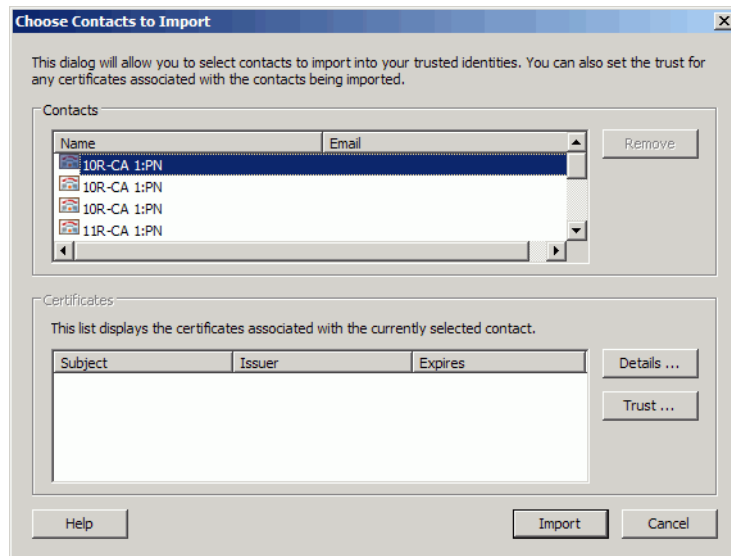


2. If the FDF file is signed, the signature can be validated, AND a trust level has been specified by the sender, check or uncheck **Accept the level of Trust specified by the signer for all Contacts in this file**.

Note: The box is disabled if either of the above conditions are not met. If the FDF is signed by someone you trust but their signature has a status of UNKNOWN, you may be able to simply add the sender to your list of trusted identities. To do so, choose **Signature Properties > Show Certificate > select the Trust tab > and choose Add to Trusted Identities**.

- If the checkbox is selected, all contacts associated with this certificate will receive the level of trust that was set by the user that signed the FDF file.
 - If the checkbox is not selected, no trust level will be set for these certificates. The certificate cannot be used for many actions (such as providing a valid timestamp or encrypting) until a trust level is set as described in the user documentation.
3. Choose **Add Contacts to List of Trusted Identities**.
 4. If there are multiple contacts in the file, the Choose Contacts to Import dialog appears. Remove those that are not wanted and highlight the rest.
 5. Choose **Import**.
 6. Choose **OK** in the confirmation dialog.

Figure 130 Making a contact a trusted identity



10.2.3.4 Importing Timestamp Server Settings

In enterprise settings, servers do not usually have to be manually configured. Timestamp server administrators often export the server information to an FDF file which is emailed or made available on a network. Users can import (add) directory server settings through the Security Settings user interface or simply by double clicking on the FDF file containing the data.

To import the server settings:

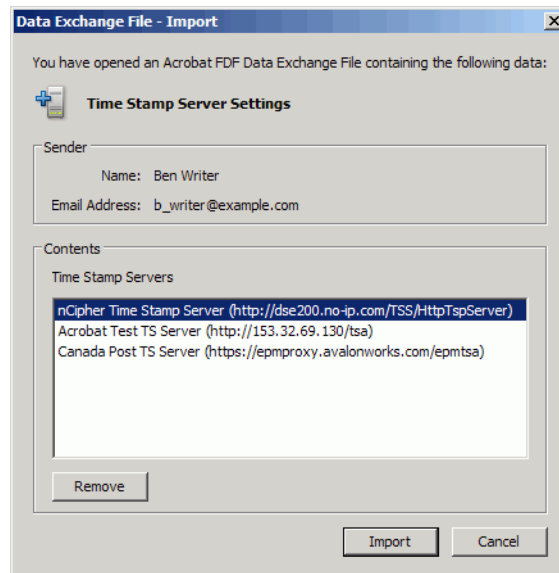
1. Locate the FDF file: find the file in an email or on the local file system and double click on it.

The FDF can also be imported through the Security Settings Console by choosing **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Security Settings**, selecting **Time Stamp Servers** in the left-hand list, and choosing **Import**.

2. Review the sender's details. Note the following:

- If the FDF is unsigned, no Signature panel appears in the import dialog.
- If the FDF is signed, you can use the **Signature Properties** button to find out more information about the sender and the validity of the signature.

Figure 131 Timestamps: Importing server details from an FDF file



3. Review the timestamp server list. Note the following behavior:

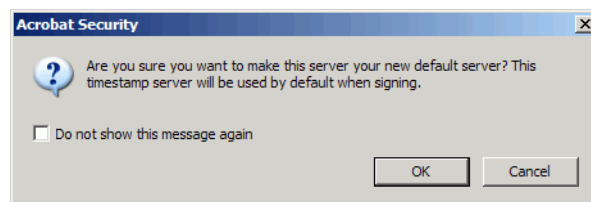
- If there is more than one server listed, all of the servers will be imported even though only one is highlighted.
- At import time, you will be asked if you want to make the highlighted server the default server.

Note: If there is more than one server and you do not want to import all of them, highlight those that should not be imported and select **Remove**.

4. Choose **Import**.

A dialog appears asking if the first (or only) server in the server list should be used as the default.

Figure 132 Timestamps: Importing a server



5. Choose **Yes** or **No**.

If **No** is selected, a default timestamp server must be set before timestamps can be used. To set a default timestamp server, Choose **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Security Settings** > **Time Stamp Servers**, select a server, and choose **Set Default**.

6. After the import completes, choose **OK**.

The settings are automatically imported and should now appear in your list of Time Stamp Servers.

10.2.3.5 Importing Directory Server Settings

In enterprise environments, administrators often set up user machines or export the configuration details to an FDF file which is emailed or made available on a network. In the latter case, you can import the server settings through the Security Settings Console or simply by double clicking on the FDF file containing the data.

To add server settings from a file:

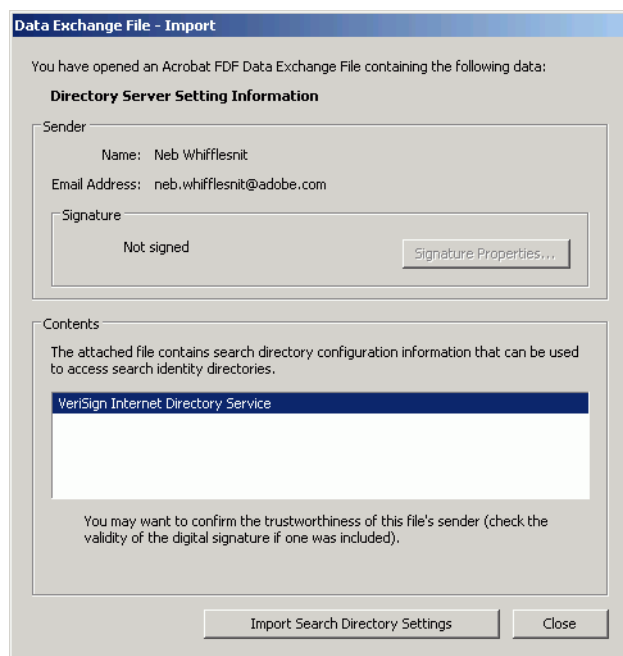
1. Locate the FDF file: find the file in an email or on the local file system and double click on it.

The FDF can also be imported through the Security Settings Console by choosing **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Security Settings**, selecting **Directory Servers** in the left-hand list, and choosing **Import**.

2. Review the sender's details. Verify the signature properties if needed (Figure 133).

Note: If the FDF is unsigned, the Signature panel will display *Not signed* and the **Signature Properties** button will be disabled.

Figure 133 Digital ID Directory servers: Importing



3. Choose **Import Search Directory Settings**.

4. If a confirmation dialog appears, choose **OK**.

This dialog will not appear if **Do not show this message again** was previously selected.

5. Choose **Close**.

The settings are automatically imported and should now appear in the Directory Servers list in the Security Settings Console.

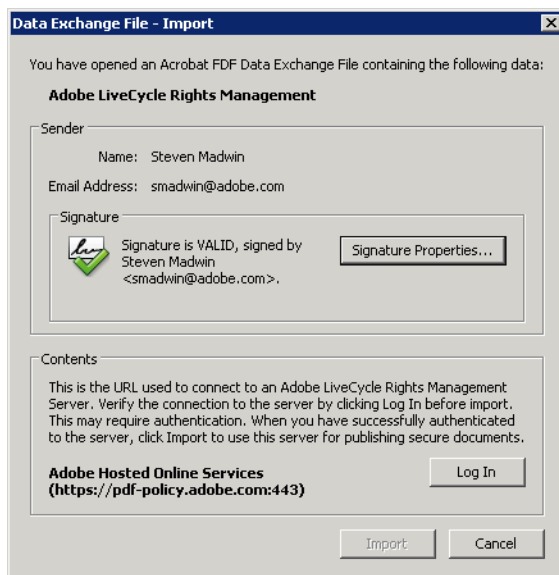
10.2.3.6 Importing Adobe LiveCycle Rights Management Server Settings

In enterprise settings, administrators often set up user machines or export the configuration details to an FDF file which is emailed or made available on a network. In the latter case, you can import the server settings through the Security Settings Console or simply by double clicking on the FDF file containing the data.

To import the server settings:

1. Locate the FDF file: find the file in an email or on the local file system and double click on it.
The FDF can also be imported through the Security Settings Console by choosing **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Security Settings**, selecting **Adobe LiveCycle Rights Management Servers** in the left-hand list, and choosing **Import**.
2. Review the sender's details. Note the following:
 - If the FDF is unsigned, no Signature panel appears in the import dialog.
 - If the FDF is signed, you can use the **Signature Properties** button to find out more information about the sender and the validity of the signature.

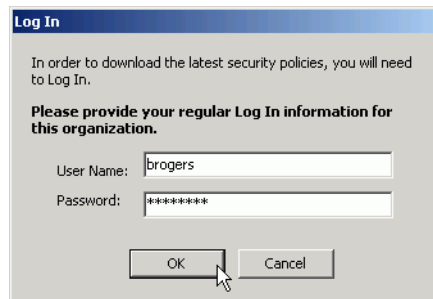
Figure 134 Importing Adobe LiveCycle Server settings



3. Choose **Log In**.

Tip: You must identify yourself to the server before you will be allowed to import these settings. The Import button does is disabled until you log in.

Figure 135 Logging in to an Adobe LiveCycle Rights Management Server



4. Choose **OK**.
5. Choose **Import**.
6. If you do not already have a default Adobe LiveCycle Rights Management Server, a dialog appears asking whether or not you want to make this your default server, choose **Yes** or **No**.
7. Choose **OK**.

The settings are automatically imported and should now appear in the Adobe LiveCycle Rights Management Servers list in the Security Settings Console.

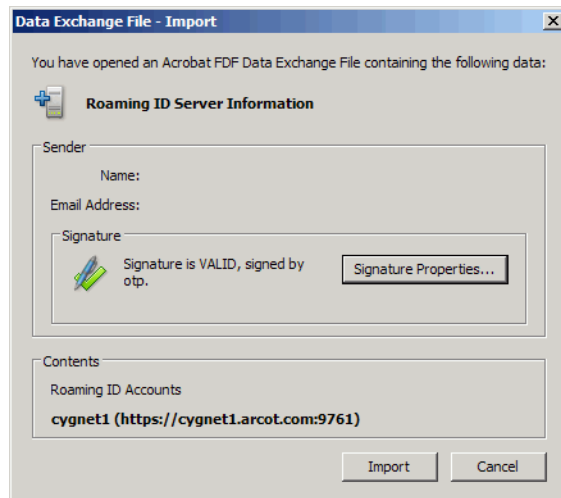
10.2.3.7 Importing Roaming ID Account Settings

In enterprise settings, administrators often set up user machines or export the configuration details to an FDF file which is emailed or made available on a network. In the latter case, you can import the server settings through the Security Settings Console or simply by double clicking on the FDF file containing the data.

To import the server settings:

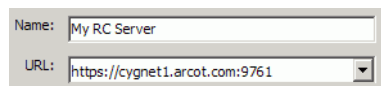
1. Locate the FDF file: find the file in an email or on the local file system and double click on it.
The FDF can also be imported through the Security Settings Console by choosing **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Security Settings**, selecting **Roaming ID Accounts** in the left-hand list, and choosing **Import**.
2. Review the sender's details. Note the following:
 - If the FDF is unsigned, no Signature panel appears in the import dialog.
 - If the FDF is signed, you can use the **Signature Properties** button to find out more information about the sender and the validity of the signature.

Figure 136 Importing roaming ID server settings



3. Choose **Import**.
4. Verify the roaming ID account name and server URL.

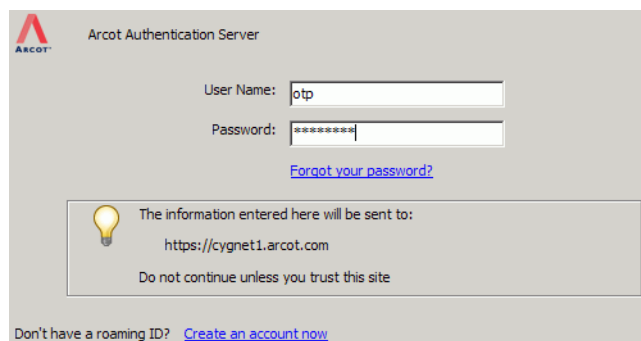
Figure 137 Roaming ID server name and URL



5. Choose **Next**.
6. Enter a user name and password.

Tip: The topmost portion of this dialog is customizable and server-dependant. The fields will remain the same, but the branding will vary.

Figure 138 Logging in to a roaming ID server



7. Choose **Next**.
8. After the confirmation that you have downloaded the roaming ID(s) appears, choose **Finish**.
The server settings and associated certificates are automatically imported and will now appear in the Roaming ID Accounts list in the Security Settings Console.

Figure 139 Downloaded roaming ID certificates

You have downloaded the following roaming ID(s):

Name	Issuer	Expires
otp	arcot	2007.08.25 23:17:33 Z

10.2.3.8 Importing a Trust Anchor and Setting Trust

Users occasionally need to import a trust anchor so that certificates that chain up to that anchor will also be trusted. This is particularly true in large organizations, and system administrators often distribute a trust anchor so that everyone within that organization can trust everyone else at the same level for signature workflows. For more information about trust anchors, see [“Distributing a Trust Anchor or Trust Root” on page 155](#).

To import a certificate that will be used as a trust anchor:

1. Open the FDF with one of the following methods:

- Click on the FDF file. It may be an email attachment or a file on a network or your local system.
- In Acrobat or Adobe Reader choose **File > Open**, browse to the FDF file, and choose **Open**.

Note: It is unlikely that you will receive a signed FDF file containing a trusted root. However, if you do, simply check **Accept the level of trust specified by the signer for all contacts in this file** and then choose **Close**. Skip the rest of the steps.

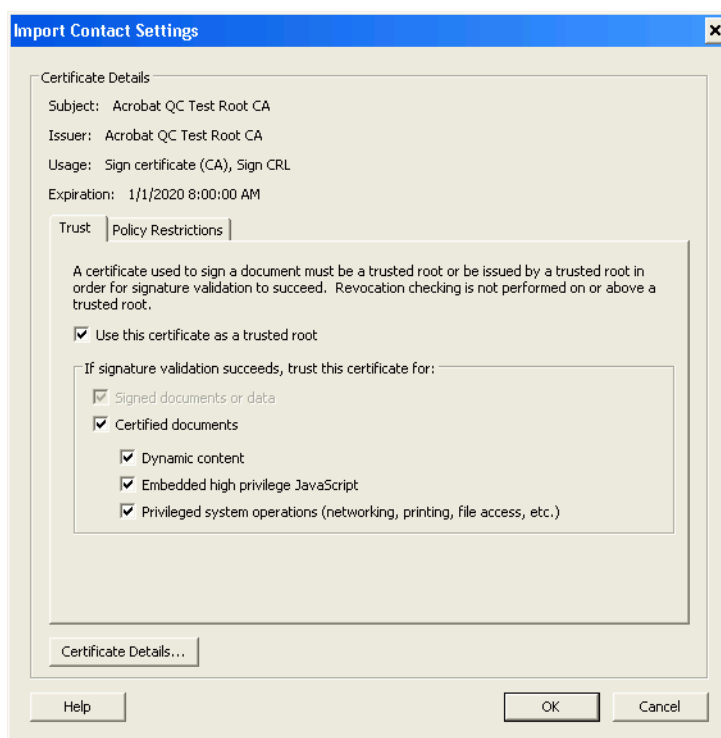
2. For unsigned FDF files containing a trusted root (the most likely case), choose **Set Contact Trust**.
3. Import the certificates.
4. Do one of the following:
 - If you already have the certificate:
 1. Choose **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Manage Trusted Identities**.
 2. Choose **Certificates** in the **Display** drop down list.
 3. Select the certificate.
 4. Choose **Edit Trust**.
 - If the certificate is in a signature:
 1. Right click and choose **Signature Properties**.
 2. Choose **Show Certificate**.
 3. Select the Trust tab.
 4. Choose **Add to Trusted Identities**.

Tip: If **Add to Trusted Identities** is disabled, the identity is already on your Trusted Identities list. To change the trust settings, you must use the first method above.

5. On the Trust tab, select the trust options. In enterprise settings, an administrator should tell you which trust settings to use.

Note: During an import action, recipients of the distributed trust anchor may be able to inherit its trust settings. Once you've verified the sender, you usually want to accept these settings so you can use the certificate they way the sender intended.

Figure 140 Certificate trust settings



- **Use this certificate as a trusted root:** Makes the certificate a trust anchor. The net result is that any certificates which chain up to this one will also be trusted for signing. At least one certificate in the chain (and preferably only one) must be a trusted root (trust anchor) to validate signatures and timestamps.

Tip: There is no need to make end entity certificates trust anchors if they issued by a certificate holder whose certificate you have configured as a trust anchor. It is best practice to trust the topmost certificate that is reasonable to trust because revocation checking occurs on every certificate in a chain until that anchor is reached. For example, in a large organization, it is likely you would want to trust your company's certificate. If that certificate was issued by VeriSign, you would not want to make VeriSign a trusted root unless you wanted to trust every certificate that chains up to VeriSign.

- **Signed documents or data:** Trusts the certificate for approval signatures.

Tip: This setting is disabled because if the certificate is set as a trust anchor. Trust anchors are automatically trusted for approval signatures.

- **Certified documents:** Trusts the certificate for certification signatures.

- **Dynamic content:** Trusts multimedia and other dynamic content in certified documents. Selecting this option automatically adds documents that are certified with this certificate to the Trusted Documents list which is maintained by the Multimedia Trust Manager. For this reason, verify your application environment is configured correctly. For details, [“Controlling Multimedia” on page 136](#).
- **Embedded high privilege JavaScript:** Trusts embedded scripts. Certificate settings do not override application-level settings, so even if JavaScript is enabled for a particular certificate, it may not execute unless the application’s preferences allow it. This option requires that the application environment be configured correctly. For details, see [“Setting JavaScript Options” on page 139](#).
- **Privileged system operations (networking, printing, file access, etc.):** Some operations represent a security risk more serious than others. Acrobat considers the following operations potential threats to a secure application operating environment: Internet connections, cross domain scripting, silent printing, external-object references, and FDF data injection. If this checkbox is checked, documents that are certified with this certificate will allow these actions.

Tip: This feature interacts with the Enhanced Security preferences which may be set by choosing **Edit > Preferences > Security (Enhanced)**. The application always takes the most permissive setting when determining what is allowed. For example, if the trust level for this certificate does not allow privileged operations but the certified file resided in a privileged location, then these operations will be permitted.

6. If you need to specify a policy restriction, do so. Most users only need to set policy restrictions at the request of their administrator. [“Certificate Trust Settings” on page 35](#).
7. Choose **OK** twice.
8. Choose **Close**.

Table 5 Security Terms

.apf	See Adobe Profile Files.
.cer	Certificate format: 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.
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.
ALCRMS	Adobe LiveCycle Rights Management Server.
approval signature	A signature used to indicate approval of, or consent on, the document terms.
ASPKI	Acrobat's Public Key Infrastructure Library (ASPKI) is 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.
CA	See certificate authority.
CDS	See Certified Document Services.
CDS digital ID	A digital ID issued by a certified document services provider.
CDS digital ID certificate	See CDS digital ID.
certificate authority (CA)	An entity that issues trusted roots.
certificates	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.
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)	A joint solution offered by Adobe and its security partners that can help recipients trust a PDF document. CDS can help provide assurance of the author's identity while also showing that the PDF document has not been modified. CDS is the only security solution that provides automatic validation of these attributes in Adobe Reader or Acrobat without also requiring additional software or configuration changes by the recipients.
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.
CRL	See Certificate Revocation List.

Table 5 Security Terms

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.
CSP	See Cryptographic Service Provider
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
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.
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.
EE	See end entity certificate.
electronic signatures	A digital signature.
embedded JavaScript	JavaScript that exists within a document rather than that which is executed from the JavaScript Console or through a batch process.
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 cancelled (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 bottom-most and end user certificate in a certificate chain is called an "end entity" (EE) certificate. It is the certificate that the holder uses for signing and others use for certificate encryption.
GeoTrust	An Adobe security partner that has joined the Adobe CDS program to provide CDS digital IDs to end users and organizations. As of Acrobat 6, Adobe Reader and Acrobat trust CDS digital IDs and are able to validate signatures that use GeoTrust digital IDs, without requiring any special application configuration.
ICA	See intermediate certificate authority.
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.
intermediate certificate authority (ICA)	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.
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.

Table 5 Security Terms

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.
OCSF	See Online Certificate Status Protocol.
Online Certificate Status Protocol (OCSF)	OCSF defines a protocol for determining the revocation status of a digital certificate without requiring a CRL. Unlike CRL, OCSF obviates the need to frequently download updates to keep certification status lists current. Acrobat's OCSF revocation checker adheres to RFC 2560.
organization digital ID, desktop	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 to perform signing operations, at the desktop, on behalf of the company.
organization digital ID, server	A digital ID issued on behalf of an organization or non-human entity (e.g. Adobe Public Relations Department, Cisco Corporation, etc.) for performing server-based, automated signing operations.
PKCS	A group of Public Key Cryptography Standards authored by RSA Security
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 PKCS# 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.
PKCS#12	Personal Information Exchange Syntax Standard: 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	Certificate Message Syntax (CMS): 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.
Policy Server	As of Acrobat 9, Adobe Policy Server is renamed to Adobe LiveCycle Rights Management Server
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.
qualified certificates	A qualified certificate 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 German digital signature law, and most qualified certificates currently originate from German trust centers.
qualified electronic signatures	Electronic signatures that use a qualified certificate valid at the time of their creation and that have been produced with a secure signature-creation device.
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.
root certificate	The top-most issuing certificate in a certificate chain. It is sometimes used as a trust anchor.
secure signature-creation devices	(SSCD) Software or hardware products used to store and apply signature code and that are designed for qualified electronic signatures
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 <i>JavaScript for Acrobat API Reference</i> identifies the items that have restrictions.

Table 5 Security Terms

SSCD	See Secure signature-creation devices
timestamp	The date and time that a digital signature was applied. The time stamp data is 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).
trust anchor	A certificate in a certificate chain that is trusted for selected operations. It could be an intermediate certificate authority rather than a trusted 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.

Index

-
- .ade 141
- .adp 141
- .apf 178
- .apf Digital IDs no longer supported 14
- .app 141
- .asp 141
- .bas 141
- .bat 141
- .bz 141
- .bz2 141
- .cer 142, 178
- .chm 142
- .class 142
- .cmd 142
- .com 142
- .command 142
- .cpl 142
- .crt 142
- .csh 142
- .exe 142
- .fxp 142
- .gz 142
- .hex 142
- .hlp 142
- .hqx 142
- .hta 142
- .inf 142
- .ini 142
- .ins 142
- .isp 142
- .its 142
- .jar 142
- .job 142
- .js 142
- .jse 142
- .ksh 142
- .lnk 142
- .lzh 142
- .mad 142
- .maf 142
- .mag 142
- .mam 142
- .maq 142
- .mar 142
- .mas 142
- .mat 142
- .mau 143
- .mav 143
- .maw 143
- .mda 143
- .mde 143
- .mdt 143
- .mdw 143
- .mdz 143
- .msc 143
- .msi 143
- .msp 143
- .mst 143
- .ocx 143
- .ops 143
- .p12 178
- .p7b 178
- .p7c 178
- .pcd 143
- .pfx 178
- .pif 143
- .pkg 143
- .prf 143
- .prg 143
- .pst 143
- .rar 143
- .reg 143
- .scf 143
- .scr 143
- .sct 143
- .sea 143
- .shb 143
- .shs 143
- .sit 143
- .tar 143
- .tgz 143
- .tmp 143
- .url 143
- .vb 143
- .vbe 143
- .vbs 143
- .vsmacros 144
- .vss 144
- .vst 144
- .vsw 144
- .webloc 144
- .ws 144
- .wsc 144
- .wsf 144
- .wsh 144
- .zip 144
- .zlo 144
- .zoo 144
- 1**
- 1000 129
- 1001 129
- 1002 129
- 1003 130
- 1004 130

1005 130
1006 130
1007 130
1008 130
1009 130

2

2004 130
2006 130
2007 130
2009 130
2010 130
2011 130
2012 130
2013 130
2014 130

3

3000 131
3001 131
3002 131
3003 131

4

4000 131
4001 131
4002 131

A

Access (Microsoft) 142
Access Add-in (Microsoft), MDA Access 2 Workgroup (Microsoft) 143
Access Add-in Data (Microsoft) 143
Access Data Access Page (Microsoft) 143
Access Diagram Shortcut (Microsoft) 142
Access Macro Shortcut (Microsoft) 142
Access MDE Database file (Microsoft) 143
Access Module Shortcut (Microsoft) 142
Access Project (Microsoft) 141
Access Project Extension (Microsoft) 141
Access Query Shortcut (Microsoft) 142
Access Report Shortcut (Microsoft) 142
Access Stored Procedures (Microsoft) 142
Access Table Shortcut (Microsoft) 142
Access Type 39
Access View Shortcut (Microsoft) 143
Access Wizard Template (Microsoft) 143
Access Workgroup Information (Microsoft) 143
Actions that can be associated with a signature field 65
Active Server Page 141
adbe.pkcs7.detached 78
adbe.pkcs7.sha1 78
adbe.x509.rsa_sha1 78
Add Digital ID dialog 14
Adding a Digital ID from a PKCS#12 File 19
Adding an ID that Resides on External Hardware 27
Adding and Removing Digital ID Files from the File List 20
Adding Custom Signing Reasons 75

Adding Files to the Black and White Lists 144
Adding Someone to Your Trusted Identity List 32
Adobe Profile Files 178
Adobe Trusted Identity Updates 140
ALCRMS 178
Allowing and Blocking Specific Web Sites 147
Allowing Attachments to Launch Applications 145
Allowing Signing Reason 50
Alternate certificate URL seed value 86
An early compressed file format 144
appearanceFilter 70
Approval Signature 90
approval signature 178
Arranging Signature Fields 60
Associating a Certificate with a Contact 43
Attachment panel in Trust Manager 145
Authenticity Verification 101
Authoring a Document with Multiple Fields 62
Authoring Form Fields 93
Authoring Signable Documents 46
Authoring Signable Forms 62
Automatic updates 141
Automating signing tasks 87

B

BASIC Source Code 141
Batch Processing 141
Before You Sign ... 89
Best Practices for Signed Documents that will Change 46
Black Lists and White Lists 141
Blocked URL alert 147
Bzip 2 UNIX Compressed file (replaces BZ) 141
Bzip UNIX Compressed file 141

C

CA 178
CDS 178
CDS digital ID 178
CDS digital ID certificate 178
certificate authority (CA) 178
Certificate Chains and Trust Anchors /Roots 156
Certificate file 142
Certificate issuer and subject seed value 83
Certificate key usage seed value 84
Certificate policy seed value 85
Certificate Revocation List (CRL) 179
Certificate Revocation List. 178
Certificate Trust Settings 35
Certificate trust settings 36, 37, 176
Certificate Viewer 118
 Trust tab 33
Certificate viewer
 Trust tab 109
Certificate Viewer information 116
Certificates
 Contact Information 166
 Selecting a digital ID for export 161
 Verifying originator 119
certificates 178

- Certificates in the Trusted Identities list 158
- Certification Signature 89
- certification signature 178
- Certification Workflow for Documents with Multiple Signers 92
- certified document 178
- Certified document indicators 92
- Certified Document Services (CDS) 178
- Certified Document Services. 178
- certify or certifying 178
- Certifying a document
 - Document integrity warnings 95
 - Signature 95
- Certifying a Document is Prevented 125
- Certifying a Dynamic Form 95
- certspec 68, 69
- Changes Across Releases 68
- Changes in FDF Behavior 134
- Changing a PKCS#12 File's Password Timeout 21
- Changing a Trusted Identity's Certificate Association 44
- Changing an ID File's Password 20
- Changing Passwords 28
- Changing the Default Field Appearance 60
- Changing the Default Signing Method 48
- Check revocation 116
- Checking Certificate Revocation Status 119
- Clearing One or More Signatures 99
- Command 142
- Comment or form field may silently change 129, 130
- Compare
 - By page 124
 - By page summary report 124
- Comparing a Signed Version to the Current Version 123
- Compiled HTML Help 142
- Compressed archive (LH ARC) 142
- Compressed Archive file 144
- Compressed archive of Mac files (Stuffit) 143
- Configuring Acrobat to use a Timestamp Server 56
- Configuring Multimedia Trust Preferences 137
- Contacts
 - Deleting 45
 - Selecting certificates 43
 - Viewing details 42
- Content preview mode can suppress 128
- Content preview mode cannot suppress 128
- Controlling Access to Referenced Files and XObjects 145
- Controlling Multimedia 136
- Controlling Multimedia in Certified Documents 138
- Controlling Signing with Seed Values 67
- Creating a Blank Signature Field 57
- Creating a Custom Signature 52
- Creating a Custom Signature Appearance 53
- Creating a Custom Watermark or Background 52
- Creating a Self-Signed Digital ID 22
- Creating Multiple Copies of a Signature Field 61
- Creation time 104
- CRL 178
- Cross-domain data access 133
- Current time 104
- Custom signature appearance 52

- Custom Workflows and Beyond 86
- Customizing a Digital ID Name 17
- Customizing Field Appearances 59
- Customizing Signature Appearances 52
- Cut, Copy, and Paste Signature Fields 60

D

- Data injection 133
- Default Behavior
 - Black and White Lists 141
- Default prohibited file types 141
- Deleting a Certificate 45
- Deleting a Digital ID from the Windows Certificate Store 26
- Deleting a Directory Server 41
- Deleting a PKCS#12 Digital ID 25
- Deleting Contacts and Certificates 44
- Details 116
- digestMethod 68, 69
- Digital ID
 - Certificate viewer 18
 - Components 12
 - Configuration 24
 - Deleting 25
 - ID export options 159
 - Managing trusted identities 32
 - PKCS#12 location and password 25
- digital ID 179
- Digital ID Basics 11
- Digital ID Directory servers
 - Email details 163
 - Export destination 162
 - Importing 171
 - Sender's identify 163
 - Server list 39
 - Setting defaults 41
 - Setting server details 40
- Digital ID files
 - Password configuration 21, 28
 - Timeout settings 22
- Digital ID Files menu 19
- Digital ID format selection 23
- Digital ID Management and the Security Settings Console 14
- Digital ID Storage Mechanisms 12
- Digital ID-related file types 13
- Digital IDs
 - Searching for certificates 34
- digital signature 179
- Digital Signature Properties
 - Document Versioning panel 127
 - Modifications panel 123
- digitally sign 179
- Directory Name 39
- Displaying the Signer's Certificate 117
- Distributing a Trust Anchor or Trust Root 155
- Document Behavior After Signing 124
- Document contains hidden behavior 129, 130
- Document contains links to external PDFs 130
- Document Defensibility 91

- document integrity 179
- Document Integrity and Preview Mode 126
- Document Integrity Checks for 8.0 121
- Document Integrity Verification 101
- Document links to external content 131
- Document Locking 91
- Document may not open in the future 130
- Document may silently launch menu items 130
- Document Message Bar
 - No dynamic content message 129
 - Suppressible rich content 128
- Document Status Definitions 113
- Documentation related to Acrobat security 10
- DOS CP/M Command file, Command file for Windows NT 142
- Downloaded roaming ID certificates 175
- dynamic feature warnings 129
- Dynamic form certification setting 96

E

- Edit Contact dialog 42, 44
- Editing Directory Servers Details 40
- Editing or Deleting a Signature Appearance 54
- EE 179
- electronic signatures 179
- Emailing a certificate request 161
- Emailing Certificate or Contact Data 43
- Emailing Server Details 162
- Emailing Your Certificate 159
- Emailing your certificate 160, 165, 166
- embedded JavaScript 179
- embedded validation response 179
- Embedding Revocation Information in a Signature 79
- Embedding Signature Revocation Status 49
- Enabling a Warnings Comment or Legal Attestation 51
- Enabling Document Warning Review 50
- Enabling Enhanced Security 133
- Enabling JavaScript to Set Seed Values 70
- end entity certificate (EE) 179
- end entity certificate. 179
- Enhanced Security 132
- Enhanced security
 - Configuration dialog 134
- Examples of Allowed Behavior 135
- Examples of Prevented Behavior 135
- Executable Application 141
- Executable file 142
- Execute a Menu Item 65
- Exporting a Certificate Other than Yours to a File 120
- Exporting a Trust Anchor 156
- Exporting Application Settings with FDF Files 155
- Exporting Security Settings to a File 149
- Exporting Server Details 163
- Exporting Your Certificate 158
- External connection warning 148
- External Content 131
- External Content and Document Security 132
- External streams access 133

F

- FDF Files and Security 154
- filter 68, 69, 78
- Finding a Digital ID in a Windows Certificate Store File 26
- flags 68, 69, 76, 81
- Forcing a Certification Signature 71
- Forcing Signers to Use a Specific Signature Appearance 74
- Form Field Fill in, Signing, and/or Other Actions Don't Work 125
- FoxPro Compiled Source (Microsoft) 142

G

- Generic ID Operations 15
- GeoTrust 179
- Getting and Using Your Digital ID 11
- Getting Started 8
- Giving Signers the Option to Lock a Document 73
- Glossary of Security Terms 178
- Go to 3D View 65
- Go to a Page View 65
- Gzip Compressed Archive 142

H

- Hash algorithm seed value 79, 80
- Hidden 58
- Hidden but printable 58
- High Privilege JavaScript Defined 139
- How Do I Validate a Timestamp in a Signature? 111
- How Should You Use This Guide? 9
- Hypertext Application 142

I

- Identity preferences 15
- IIS Internet Communications Settings (Microsoft) 142
- IIS Internet Service Provider Settings (Microsoft) 142
- Import Form Data 65
- Importing a Certificate From a File 33
- Importing a Trust Anchor and Setting Trust 175
- Importing Adobe LiveCycle Rights Management Server Settings 172
- Importing Adobe LiveCycle Server settings 172
- Importing and Exporting Directory Server Settings 41
- Importing Application Settings with FDF Files 164
- Importing digital ID data 33
- Importing Directory Server Settings 171
- Importing Multiple Certificates 167
- Importing multiple certificates 168
- Importing Roaming ID Account Settings 173
- Importing roaming ID server settings 174
- Importing Security Settings from a File 150
- Importing Security Settings from a Server 152
- Importing Someone's Certificate 166
- Importing Timestamp Server Settings 169
- individual digital ID 179
- Information or Setup file 142
- Initialization/Configuration file 142
- Interaction with Trust Manager 136
- intermediate certificate authority (ICA) 179

Internal Document Signature components 101
Internet access panel 146
Internet Document Set, International Translation 142
Internet Location 143
Internet Security Certificate file (MIME x-x509-ca-cert) 142
Internet URL Access 146
issuer 81

J

Java Archive 142
Java Class file 142
Javascript and Certified Documents 139
JavaScript and Dynamic Content Won't Run 125
JavaScript Security option 140
JavaScript Source Code 142
JScript Encoded Script file 142

K

keyUsage 81

L

Launch Attachment dialog 144
Legal Attestations and Warnings Comments 92
Legal Notice 117
legalAttestations 68, 69
LiveCycle Dynamic Forms and the Warning Triangle 121
Local Time versus Timestamp Time 110
lockDocument 70
Locking Fields Automatically After Signing 62
Logging in to a Device 28
Logging in to a Digital ID File 19
Logging in to a roaming ID server 174
Logging in to an Adobe LiveCycle Rights Management Server 173
Logging in to PKCS#12 Files 22

M

Mac OS Command Line executable 142
Mac OS Finder Internet Location 144
Mac OS X Installer Package 143
Macintosh BinHex 2.0 file 142
Macintosh BinHex 4 Compressed Archive 142
Make Privileged Folder Locations Recursive 136
Making a contact a trusted identity 169
Making a Field a Required Part of a Workflow 63
Manage Internet Access dialog 147
Manage Trust for Multimedia Content dialog 136
Manage Trusted Identities menu item 32
Managing Certificate Trust and Trusted Identities 30
Managing Contacts 42
Managing PKCS#12 Digital ID Files 19
Managing Windows Digital IDs 26
Manually Configuring a Directory Server 39
Maximum Number of Records to Receive 40
mdp 68, 69
Media Attachment Unit 143
Message digest 179
message digest 101

Microsoft Management Console Snap-in Control file (Microsoft) 143
Microsoft Object Linking and Embedding (OLE) Control Extension 143
Migrating and Sharing Security Settings 149
Mouse Down 65
Mouse Enter 65
Mouse Exit 65
Mouse Up 65
MS Exchange Address Book file, Outlook Personal Folder file (Microsoft) 143
Multimedia behavior workflow 137
Multimedia Trust (legacy) 137

N

Name 58
No external dependencies or dynamic content 129

O

OCSP 180
Office Profile Settings file 143
oid 81
On Blur 65
On Focus 65
Online Certificate Status Protocol (OCSP) 180
Open a File 66
Open a Web Link 66
organization digital ID, desktop 180
organization digital ID, server 180
Orientation 58

P

Page content may silently change 130, 131
Password 40
PDF content contains errors 131
PDF Content with variable rendering 130
PDF Signature Report
 Content which cannot be suppressed in preview mode 128
 Suppressed content 129
PDF Signature Reports 127
PDF/X-5 145
Personalizing an ID name 17
PKCS 180
PKCS#11 device 180
PKCS#11 digital ID 180
PKCS#11 format 180
PKCS#11 module 180
PKCS#11 Security Settings menu items 27, 28, 29
PKCS#11 token 180
PKCS#12 180
PKCS#7 180
Play a Sound 66
Play Media (Acrobat 5 Compatible) 66
Play Media (Acrobat 6 Compatible) 66
Policies 116
Policy OID 85
Policy Server 180

- Port 40
- Presentation elements may change appearance 130
- Preventing Multimedia Playback in Certified Documents 138
- Preview document mode preference 48
- Preview Mode and Signing Workflows 126
- Preview Mode and Validation (View Signed Version) 127
- privileged context 180
- Problems encountered 116
- Program file 143
- Providing Instructions to the Trusted Root Recipients 158

Q

- qualified certificates 180
- qualified electronic signatures 180

R

- Read an Article 66
- Read Only 58
- Reason field behavior 75
- reasons 68, 70
- Registering a Digital ID for Use in Acrobat 13
- Registration Information/Key for Windows 95/98, Registry Data file 143
- Requesting a Certificate via Email 161
- Requesting a Digital ID via Email 33
- Required 58
- Required field not signed alert 64
- Requiring Document Warning Review Prior to Signing 51
- Requiring Preview Mode 47
- Reset a Form 66
- Resetting the Black and White Lists 144
- Resource access 146
- Responding to an Email Request for a Digital ID 164
- Restricting Signing to a Roaming ID 86
- Revalidate signatures warning 112
- Revocation 116
- Roadmap to Other Security Documentation 9
- roaming ID 180
- Roaming ID seed value 86
- Roaming ID server name and URL 174
- root certificate 180
- Rules for opening a PDF via FDF 134, 154
- Run a JavaScript 66

S

- Saving Certificate or Contact Details to a File 43
- Saving Your Digital ID Certificate to a File 160
- Script injection 133
- Search Base 40
- Searching for a document recipients 35
- Searching for Digital ID Certificates 34
- secure signature-creation devices 180
- Secure time 104
- security restricted property or method 180
- Security setting import
 - Success dialog 152
- Security Setting Import and Export 149

- Security setting preferences for server import 152
- Security settings
 - Document message bar 151
 - Encryption method 150
 - Export dialog 150
 - Import from a file panel 151
- Security Settings Console 14
- Security settings menu and manager 14
- Security Settings menu items 162
- Security Terms 178
- Seed value
 - Custom signing reason 76
 - Forcing mdp selection during certification 72
 - lockDocument 74
 - mdp 73
 - Reason not allowed error 76
 - signatureAppearance 75
 - Specifying certificates for signing 80
 - Specifying signature components 78
- Seed Value Basics 67
- Seed values
 - certSpec properties 81
 - Changes across releases 68
 - Custom legal attestations 73
 - JavaScript debugger 71
 - Object properties and descriptions 69
 - timeStampSpec properties 76
- Selecting a certificate chain for export 157
- Selecting a digital ID 165
- Self-expanding archive (used by Stuffit for Mac files and possibly by others) 143
- Server Name 39
- Set Layer Visibility 66
- Setting Digital Signature Validation Preferences 103
- Setting Identity Information 14
- Setting JavaScript Options 139
- Setting Signing Preferences 47
- Setting the Certificate Trust Level 158
- Setting up a Document for Certification 93
- Setting up the Signing Environment 46
- Setting up Your Environment for Signature Validation 102
- Sharing (Exporting) a Digital ID Certificate 16
- Sharing Settings & Certificates with FDF 152
- Shell Scrap Object file 143
- shouldAddRevInfo 68, 70
- Show/Hide a Field 66
- Showing Location and Contact Details 50
- Sign Document dialog
 - With Lock Document checkbox added 74
- Signature appearance
 - Configuration 54
 - New button 53
- Signature creation preferences 49
- Signature field
 - Action properties 65
 - Appearance properties 59
 - Default appearance 57
 - Edit options 60
 - General properties 59
 - Multiple copy options 61

- Signing properties 63
- Signature field sign menu 97
- Signature Properties
 - Summary 108
- Signature Report Error Codes 129
- Signature status cheat sheet 114
- Signature Status Definitions 113
- Signature Types 89
- Signature validation confirmation 107
- Signature Validity Basics 100
- Signature verification preferences 103
- Signatures tab
 - Validate signature 107
- Signer Details 116
- Signing a document
 - Signature details 98
- Signing an FDF file 155
- Signing Basics 89
- Signing Documents 89
- Signing Documents in Acrobat 96
- Signing environment preferences 47
- Signing in a Browser 98
- Signing User Interface 90
- Signing With a Certification Signature 90
- Signing with an Approval Signature 96
- Silent printing 133
- Specifying a Default Directory Server 41
- Specifying a Post-Signing Action 64
- Specifying a Signature Hash Algorithm 79
- Specifying a URL When a Valid Certificate is not Found 85
- Specifying Alternate Signature Handlers and Formats 77
- Specifying Certificate Properties for Signing 80
- Specifying Certificates by Key Usage 83
- Specifying Certificates by Policy 84
- Specifying Digital ID Usage 15
- Specifying General Field Properties 58
- Specifying Signing Certificates Origin 82
- Specifying Timestamps for Signing 76
- SSCD 181
- Status Icons and Their Meaning 113
- subFilter 68, 70
- subfilter 78
- subject 82
- subjectDN 82
- Submit a Form 66
- Summary 116
- Supported Seed Values 69

T

- Tape Archive file 143
- Temporary file or Folder 143
- Text appearance may silently change 130
- The document contains a dynamic form 130
- This server requires me to log on 40
- Time stamp server error 77
- Timeout 40
- timestamp 181
- Timestamp server seed value 77
- Timestamps

- Date/Time tab 111
- Entering server details 56
- Importing a server 170
- Importing server details from an FDF file 170
- Local, machine time 55, 110
- Trusted stamp 55, 111
- Untrusted stamp 55, 111
- timeStampspec 68, 70
- Tooltip 58
- Troubleshooting a Document Integrity Problem 120
- Troubleshooting a Signature or Document Status 115
- Troubleshooting an Identity Problem 115
- Troubleshooting Digital ID Certificates 116
- Trust 116
- trust anchor 181
- Trusted Identities
 - Viewing revocation status 120
- Trusted identities 12
- Trusting certificate from a document warning 109
- Trusting Windows root certificates 105
- Turning Internet Access Off and On 146

U

- Uncategorized warnings 131
- UNIX csh shell script 142
- UNIX ksh shell script 142
- UNIX Tar file Gzipped 143
- Unlocking a Field Locked by a Signature 66
- Unrecognized PDF content 131
- Untrusted signature 36
- url 76, 82
- urlType 82
- Usage options for a digital ID 16
- User name 40
- Using Directory Servers to Add Trusted Identities 38
- Using Root Certificates in the Windows Certificate Store 104
- Using Seed Values to Individual Form Fields 93
- Using Timestamps During Signing 55

V

- Validate all signatures dialog 107
- Validating a Single Signature in Acrobat 106
- Validating All Signatures in Acrobat 107
- Validating an Problematic Signature (trusting a signer on-the-fly) 108
- Validating Signature Timestamps 110
- Validating Signatures 100
- Validating Signatures Automatically 102
- Validating Signatures for other Document Versions 110
- Validating Signatures Manually 106
- Validating Signatures with Adobe Reader 106
- Validating Signatures with Timestamps and Certificate Policies 105
- VBScript Encoded Script file 143
- VBScript file or Any VisualBasic Source 143
- VBScript Script file, Visual Basic for Applications Script 143
- Verifying the Identity of Self-Signed Certificates 118
- version 68, 70
- Viewing a List of Post-Signing Modifications 122

- Viewing All of Your Digital IDs 16
- Viewing and Comparing Changes and Versions 122
- Viewing and Editing Contact Details 42
- Viewing Digital ID Certificates in the Certificate Viewer 17
- Visible 58
- Visible but doesn't print 58
- Visio Stencil (Microsoft) 144
- Visio Template (Microsoft) 144
- Visio Workspace file (Microsoft) 144
- Visual Studio .NET Binary-based Macro Project (Microsoft) 144
- Visual Test (Microsoft) 143

W

- Weblink 133
- What is a Digital ID? 11
- What is a timestamp? 111
- What is a Trusted Identity? 30
- What is Trust? 30
- What Makes a Signature Valid? 100
- What's in this Guide? 8
- When Timestamps Can't be Verified... 112
- Who Should Read This Guide? 8
- Why Attach a File that's on the Black List? 141
- Why Can't I Certify? 96
- Windows Control Panel Extension (Microsoft) 142
- Windows digital ID menu 26
- Windows Explorer Command 143
- Windows Help file 142
- Windows Installer file (Microsoft) 143
- Windows Installer Patch 143

- Windows Program Information file (Microsoft) 143
- Windows Screen Saver 143
- Windows Script Component 144
- Windows Script Component, Foxpro Screen (Microsoft) 143
- Windows Script file 144
- Windows Script Host Settings file 144
- Windows SDK Setup Transform Script 143
- Windows Shortcut file 142
- Windows Shortcut into a Document 143
- Windows System file 143
- Windows Task Scheduler Task Object 142
- WinRAR Compressed Archive 143
- Working with Attachments 141
- Working with Signature Fields 56

Y

- You can customize the way a certified document behaves for signers by giving form fields additional features with seed values. For example, you can preconfigure custom signing reasons or limit signing to only those with certificates with predefined characteristics. Certifying a Document 93
- Your server may require additional or different authentication steps. Follow directions that appear in the dialogs. Managing IDs Stored on Hardware Devices 27

Z

- ZoneLabs ZoneAlarm Mailsafe Renamed .PIF file 144