3.3 Certificate Trust Settings ................................................................................................................36
  3.3.1 Setting Certificate Trust ..................................................................................................................37
  3.3.2 Setting Certificate Policy Restrictions ...............................................................................................39
  3.3.3 Using Certificates for Certificate Security (Encryption) .................................................................40

3.4 Using Directory Servers to Add Trusted Identities .............................................................................40
  3.4.1 Manually Configuring a Directory Server .......................................................................................41
  3.4.2 Editing Directory Servers Details ..................................................................................................42
  3.4.3 Deleting a Directory Server ............................................................................................................42
  3.4.4 Specifying a Default Directory Server ............................................................................................43
  3.4.5 Importing and Exporting Directory Server Settings .......................................................................43

3.5 Managing Contacts ...................................................................................................................................43
  3.5.1 Viewing and Editing Contact Details ..............................................................................................43
  3.5.2 Emailing Certificate or Contact Data ...............................................................................................44
  3.5.3 Saving Certificate or Contact Details to a File ..................................................................................44
  3.5.4 Associating a Certificate with a Contact ..........................................................................................45
  3.5.5 Changing a Trusted Identity’s Certificate Association ......................................................................45
  3.5.6 Deleting Contacts and Certificates ..................................................................................................46

4 Authoring Signable Documents ...............................................................................................................48

4.1 Setting up the Signing Environment ....................................................................................................48
  4.1.1 Setting Signing Preferences ............................................................................................................49
    4.1.1.1 Requiring Preview Mode .............................................................................................................49
    4.1.1.2 Changing the Default Signing Method .......................................................................................50
    4.1.1.3 Embedding Signature Revocation Status ....................................................................................50
    4.1.1.4 Allowing Signing Reasons .........................................................................................................51
    4.1.1.5 Showing Location and Contact Details .......................................................................................51
    4.1.1.6 Enabling Document Warning Review .......................................................................................51
    4.1.1.7 Requiring Document Warning Review Prior to Signing ............................................................51
    4.1.1.8 Enabling a Warnings Comment or Legal Attestation .................................................................52
  4.1.2 Custom Signature Appearances ........................................................................................................52
    4.1.2.1 Creating a Custom Signature ......................................................................................................53
    4.1.2.2 Creating a Custom Watermark or Background ..........................................................................53
    4.1.2.3 Creating a Custom Signature Appearance ..................................................................................53
    4.1.2.4 Editing or Deleting a Signature Appearance .............................................................................55
  4.1.3 Signing with a Timestamp ................................................................................................................55
  4.1.4 Changing the Default Signing Method .............................................................................................57

4.2 Working with Signature Fields ..............................................................................................................58
  4.2.1 Creating a Blank Signature Field .....................................................................................................58
  4.2.2 Specifying General Field Properties ................................................................................................59
  4.2.3 Customizing Field Appearances ......................................................................................................60
  4.2.4 Changing the Default Field Appearance ..........................................................................................61
  4.2.5 Cut, Copy, and Paste Signature Fields .............................................................................................61
  4.2.6 Arranging Signature Fields ..............................................................................................................62
  4.2.7 Creating Multiple Copies of a Signature Field ..................................................................................62

4.3 Authoring Signable Forms .....................................................................................................................63
  4.3.1 Authoring a Document with Multiple Fields ..................................................................................63
  4.3.2 Locking Fields Automatically After Signing ....................................................................................64
  4.3.3 Making a Field a Required Part of a Workflow ...............................................................................65
  4.3.4 Specifying a Post-Signing Action ......................................................................................................65
  4.3.5 Unlocking a Field Locked by a Signature ..........................................................................................67

4.4 Checking Document Integrity .................................................................................................................68
5 Signing with a Digital Signature .................................................................73
  5.1 Signing Basics............................................................................................73
    5.1.1 Before You Sign ..................................................................................73
    5.1.2 Signature Types ..................................................................................73
    5.1.3 Signing User Interface ........................................................................74
  5.2 Signing With a Certification Signature ....................................................74
    5.2.1 Certification Workflow for Documents with Multiple Signers ..........76
    5.2.2 Setting up a Document for Certification ..............................................77
    5.2.3 Certifying a Document .........................................................................78
    5.2.4 Certifying a Dynamic XFA Form ..........................................................80
    5.2.5 Why Can't I Certify? ............................................................................80
  5.3 Signing with an Approval Signature .......................................................81
    5.3.1 Signing Documents in Acrobat ..............................................................81
    5.3.2 Signing in a Browser .............................................................................83
    5.3.3 Clearing One or More Signatures .........................................................83
  6 Digital Signature Validation .......................................................................84
    6.1 Signature Validity Basics .........................................................................84
      6.1.1 What Makes a Signature Valid? ..........................................................84
        6.1.1.1 Authenticity Verification ..............................................................84
        6.1.1.2 Document Integrity Verification ................................................85
      6.1.2 Signature Status ................................................................................85
    6.1.3 Running JavaScript and Dynamic Content ..........................................87
    6.2 Setting up Your Environment for Signature Validation .........................87
      6.2.1 Setting Up Automatic Signature Validation ......................................87
      6.2.2 Setting Digital Signature Validation Preferences ............................88
      6.2.3 Using Root Certificates in the Windows Certificate Store ................89
    6.3 Validating Signatures Manually .............................................................90
      6.3.1 Validating Signatures with Adobe Reader .......................................90
      6.3.2 Validating a Single Signature in Acrobat .........................................91
      6.3.3 Validating All Signatures in Acrobat ...............................................91
      6.3.4 Validating an Unknown Signature .................................................92
      6.3.5 Validating a Signature for an Earlier Document Version ................94
      6.3.6 Validating Signature Timestamps .......................................................95
        6.3.6.1 When Timestamps Can't be Verified .........................................98
    6.4 When the Status Icon is Not a Green Check .........................................98
      6.4.1 Green Check and Caution Triangle ..................................................99
      6.4.2 Question Mark and Caution Triangle ...............................................99
      6.4.3 Question Mark and Person ...............................................................100
      6.4.4 Question Mark and Pen ...................................................................100
      6.4.5 Red X and Pen .................................................................................101
    6.5 XFA Dynamic Forms and the Warning Triangle ....................................101
    6.6 Troubleshooting Digital ID Certificates ...............................................102
      6.6.1 Displaying Someone's Certificate in the Certificate Viewer ...........102
7 Controlling Signing with Seed Values ................................................................. 112
  7.1 Seed Value Basics .......................................................................................... 112
    7.1.1 Changes Across Releases ........................................................................... 113
    7.1.2 Supported Seed Values .............................................................................. 113
    7.1.3 Enabling JavaScript to Set Seed Values ..................................................... 114
  7.2 Forcing a Certification Signature and Document Locking ................................. 115
  7.3 Adding Custom Signing Reasons ..................................................................... 117
  7.4 Specifying Timestamps for Signing ................................................................. 119
  7.5 Specifying Alternate Signature Handlers and Formats ....................................... 120
  7.6 Specifying a Signature Hash Algorithm .......................................................... 121
  7.7 Embedding Revocation Information in a Signature ........................................... 122
  7.8 Specifying Certificate Properties for Signing .................................................... 122
    7.8.1 Specifying Signing Certificates Origin ......................................................... 125
    7.8.2 Specifying Certificates by Key Usage ........................................................ 126
    7.8.3 Specifying Certificates by Policy ............................................................... 127
    7.8.4 Specifying a URL When a Valid Certificate is not Found ............................ 128
    7.8.5 Restricting Signing to a Roaming ID ............................................................ 128
  7.9 Custom Workflows and Beyond ...................................................................... 129

8 External Content and Document Security .......................................................... 132
  8.1 Controlling Multimedia .................................................................................. 132
    8.1.1 Configuring Multimedia Trust Preferences ................................................. 133
    8.1.2 Controlling Multimedia in Certified Documents ....................................... 134
  8.2 Setting High Privilege JavaScript Options ....................................................... 135
    8.2.1 High Privilege JavaScript Defined ............................................................. 135
    8.2.2 Javascript and Certified Documents .......................................................... 136
  8.3 Working with Attachments .............................................................................. 136
    8.3.1 Default Behavior: Black and White Lists ................................................... 137
    8.3.2 Adding Files to the Black and White Lists ................................................. 140
    8.3.3 Resetting the Black and White Lists ......................................................... 141
    8.3.4 Allowing Attachments to Launch Applications ........................................... 141
  8.4 Controlling Access to Content Outside a PDF .................................................. 142
  8.5 Internet URL Access ....................................................................................... 142
    8.5.1 Turning Internet Access Off and On .......................................................... 143
    8.5.2 Allowing and Blocking Specific Web Sites .................................................. 144

9 Sharing Settings & Certificates with FDF .......................................................... 145
  9.1 Importing Application Settings and Digital ID Data ......................................... 146
    9.1.1 Responding to an Emailed Request for a Digital ID .................................... 146
9.1.2 Importing Someone's Certificate ....................................................................................................................... 148
9.1.3 Importing Multiple Certificates .......................................................................................................................... 149
9.1.4 Importing Timestamp Server Settings ............................................................................................................. 150
9.1.5 Importing Directory Server Settings ................................................................................................................. 151
9.1.6 Importing Adobe Policy Server Settings ........................................................................................................... 152
9.1.7 Importing Roaming ID Account Settings ........................................................................................................ 154
9.1.8 Importing a Trust Anchor and Setting Trust .................................................................................................. 155

9.2 Exporting Application Settings and Digital ID Data ............................................................................................ 157
9.2.1 Distributing a Trust Anchor or Trust Root .......................................................................................................... 158
  9.2.1.1 Exporting a Trust Anchor ............................................................................................................................ 159
  9.2.1.2 Providing Instructions to the Trusted Root Recipients ........................................................................... 160
  9.2.1.3 Setting the Certificate Trust Level ........................................................................................................... 160
9.2.2 Exporting Your Certificate .................................................................................................................................... 161
  9.2.2.1 Emailing Your Certificate ............................................................................................................................ 161
  9.2.2.2 Saving Your Digital ID Certificate to a File ............................................................................................ 162
9.2.3 Requesting a Certificate via Email ..................................................................................................................... 163
9.2.4 Emailing Server Details .......................................................................................................................................... 164
9.2.5 Exporting Server Details ........................................................................................................................................ 166

10 Glossary of Security Terms ........................................................................................................................................ 168
11 Index ........................................................................................................................................................................... 171
1 Getting Started

1.1 What’s in this Guide?

This guide describes the digital signature features of the Acrobat 8.x 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 Usage”
- Digital signatures:
  - Chapter 4, “Authoring Signable Documents”
  - Chapter 5, “Signing with a Digital Signature”
  - Chapter 6, “Digital Signature Validation”
- Sharing certificates and server settings with data exchange files:
  - “Importing Application Settings and Digital ID Data” on page 146
  - “Exporting Application Settings and Digital ID Data” on page 157
- Securing the application environment:
  - “Controlling Multimedia” on page 132
  - “Setting High Privilege JavaScript Options” on page 135
  - “Working with Attachments” on page 136
  - “Controlling Access to Content Outside a PDF” on page 142 (only available in 7.0.5 and later)
  - “Internet URL Access” on page 142

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 Administrator 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 Usage”. In enterprise settings, the administrator may issue you an ID (or provide instructions on getting one) and may also set up your application to that it can verify (trust) signatures.

If you are a document author or manage document templates that contain signature fields, read Chapter 4, “Authoring Signable Documents”.

If you will be signing documents, configure your application and learn about the signature types and signing process as described in Chapter 5, “Signing with a Digital Signature”.

If you will be validating signatures in signed documents, configure your application and learn about what controls signature status as described in Chapter 6, “Digital Signature Validation”. If you have not previously configured a trust anchor, you may also want to read Chapter 3, “Managing Certificate Trust and Usage”.

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 8, “External Content and Document Security”.

If you need to share your certificate or server settings with someone, see Chapter 9, “Sharing Settings & Certificates with FDF”.

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 admin-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/devnet/acrobat/.

Table 1 Documentation related to Acrobat security

<table>
<thead>
<tr>
<th>Document</th>
<th>Audience</th>
<th>For information about</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acrobat SDK Documentation Roadmap</td>
<td>Developers</td>
<td>A guide to the documentation in the Adobe Acrobat SDK.</td>
</tr>
<tr>
<td>Acrobat and PDF Library API Reference</td>
<td>Developers</td>
<td>A description of the APIs for Acrobat and Adobe Reader® plug-ins, as well as for PDF Library applications.</td>
</tr>
<tr>
<td>JavaScript for Acrobat API Reference</td>
<td>Developers</td>
<td>A listing of the Acrobat JavaScript APIs.</td>
</tr>
<tr>
<td>Developing Acrobat Applications with JavaScript</td>
<td>Developers</td>
<td>Additional detail about the Acrobat JavaScript APIs.</td>
</tr>
<tr>
<td>PDF Reference 1.7</td>
<td>Developers</td>
<td>A detailed description of the PDF language.</td>
</tr>
<tr>
<td>FDF Data Exchange Specification</td>
<td>Developers</td>
<td>A object-level FDF file description. The files can be generated programmatically and used to share security-related data.</td>
</tr>
<tr>
<td>PDF Signature Build Dictionary Specification</td>
<td>Developers</td>
<td>Build properties for the PDF Reference's signature dictionary which provides interoperability details for 3rd party handlers.</td>
</tr>
<tr>
<td>Digital Signature Appearances</td>
<td>Developers &amp; administrators</td>
<td>Guidelines for creating signatures programmatically.</td>
</tr>
<tr>
<td>Guidelines for Developing CSPs for Acrobat on Windows</td>
<td>Developers &amp; administrators</td>
<td>Guidelines for developing a Cryptographic Service Provider for use with Acrobat® on the Windows® platform.</td>
</tr>
<tr>
<td>Acrobat Security Administration Guide</td>
<td>Administrators</td>
<td>Application deployment and configuration in enterprise settings.</td>
</tr>
<tr>
<td>Acrobat 8.1 Digital Signature User Guide</td>
<td>Administrators &amp; end users</td>
<td>Application usage and configuration via the user interface.</td>
</tr>
<tr>
<td>Acrobat 8.1 Security FDF User Guide</td>
<td>Administrators &amp; end users</td>
<td>A subset of the user guides that describe how to export and import security settings and certificate data with an FDF file.</td>
</tr>
<tr>
<td>Digital Signatures in the PDF Language</td>
<td>Anyone needing an overview</td>
<td>A generic description of how signature work in PDF.</td>
</tr>
<tr>
<td>Digital Signatures in Acrobat</td>
<td>Anyone needing an overview</td>
<td>A description of how signatures are implemented in Acrobat.</td>
</tr>
</tbody>
</table>
2 Getting and Using Your Digital ID

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. Digital IDs are required for using 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
- “Managing Roaming ID Accounts and IDs” on page 26
- “Managing IDs Accessible via PKCS#11 Devices” on page 28

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 include a default signature handler that can access digital IDs from any of these locations.
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 Usage”.

- **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 40.

### 2.1.2 Digital ID Related Files and Storage Mechanisms

There are several digital ID-related file types and storage mechanisms a user might encounter in signing and certificate encryption processes (Table 2). Digital IDs (the certificate and the private key) are provided to the application via digital ID service providers (sometimes called Cryptographic Service Providers or CSPs). A service provider is simply a storage mechanism 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 PKCS#11 hardware such as a smart card connected to the computer. The application remembers the data’s location when the ID is registered (imported) and is added to the Security Settings Consoles digital ID list.

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

- **PKCS#12 files**: A common file format that contains the entire digital ID and is used on both Windows and Macintosh. These files may be in the Acrobat store or the Windows store.
Windows Certificate Store: A local store that can import and export various file formats and that can be used by both Windows programs and Acrobat products.


Roaming ID servers: The private key is known only to a remote server. The server sends the certificate and its public key to users on demand. Users can import and export the certificate from Acrobat.

APF files: A legacy format that is no longer used.

### Table 2 Digital ID-related file types

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>5.x</th>
<th>6.x</th>
<th>7.x</th>
<th>8.x</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKCS#12:</td>
<td></td>
<td>Import</td>
<td>Import</td>
<td>Export</td>
<td>Export</td>
</tr>
<tr>
<td>.pfx (Win), .p12 (Mac)</td>
<td>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)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.fdf</td>
<td>An Adobe file data exchange format used for importing and exporting settings and certificates (usually PKCS#12 files).</td>
<td>Export</td>
<td>Export</td>
<td>Export</td>
<td>Export</td>
</tr>
<tr>
<td>PKCS#7:</td>
<td></td>
<td>Import</td>
<td>Import</td>
<td>Export</td>
<td>Export</td>
</tr>
<tr>
<td>.p7b, .p7c</td>
<td>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</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.cer</td>
<td>Certificate format: A Microsoft format for digital IDs usually stored in the Windows Certificate Store. Contains: Certificate and public key only</td>
<td>Export</td>
<td>Export</td>
<td>Export</td>
<td>Export</td>
</tr>
<tr>
<td>.apf</td>
<td>Adobe Profile Files (Legacy): Not used after Acrobat 5. Files can be upgraded by double clicking them. Contains: Digital ID (public and private keys)</td>
<td>Import</td>
<td>Import</td>
<td>Import</td>
<td>Import</td>
</tr>
</tbody>
</table>

### 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. When the Apply Digital Signature - Digital ID Selection dialog appears, choose **Add Digital ID**.

For more information, refer to the following:

- Finding an Existing Digital ID in a PKCS#12 File
- Finding a Digital ID in a Windows Certificate Store File
- Adding an ID that Resides on External Hardware
- Adding a Roaming ID Account to Get a Roaming ID
- Upgrading a Legacy .apf Digital ID
Upgrading a Legacy .apf Digital ID

Older application versions use a deprecated digital ID format with an .apf extension. If an .apf digital ID file is selected, users will be prompted to convert the file to a supported file type.

2.1.4 Digital ID Management

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.

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
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**

![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, do 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, you can send it to eventual document recipients 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 17
- “Customizing a Digital ID Name” on page 17
- “Viewing Digital ID Certificates in the Certificate Viewer” on page 18
2.2.1 Specifying Digital ID Usage

If a digital ID is not specified for a particular task, a prompt asks 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.

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 6).
3. Highlight an ID in the list on the right.

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.

Tip: 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 receive your certificate before:
Validating your signature. They could also trust a certificate above yours in the certificate chain. Note that a signature always include the signer’s certificate, so validation can occur with the embedded certificate if it is not already on the validator’s machine.

Encrypting a document for you using certificate security.

Certificates can be emailed or saved to a file. You can 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 161.

**Note:** To export a certificate displayed in the Certificate Viewer, choose Export on the General 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 6).
   
   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 Roaming ID Accounts, Digital ID Files, Windows Digital IDs, and 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 6).
3. Highlight an ID in the list on the right.
5. Enter a name for the ID.
2.2.5 Viewing Digital ID Certificates in the Certificate Viewer

Your personal digital IDs appear in the Security Settings Console. From there, the Certificate Viewer can be used to display the time for which the certificate is valid, usage, a unique serial number, public key method, and so on (Figure 9).

To check certificate details:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Select **Digital IDs** in the left-hand tree (Figure 6).
3. Highlight an ID in the list on the right.
4. Choose **Certificate Details**. The Certificate Viewer displays the certificate. (Figure 9). 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 certificate's trust level. If it does not already exist in the trusted identities list, the **Add to Trusted Identities** is active.
   - **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.
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.

2.3.1 Finding an Existing Digital ID in a PKCS#12 File

If a required digital ID file does not appear in the digital ID list, search for it and add it. You can browse to PKCS#12 files (.pfx or .p12) and Windows Certificate Store compatible files (.cer and .der).
Note: In enterprise settings, you may be instructed by your administrator to get a 3rd party digital ID from a specific location which provide proprietary or advanced security features. Third-party providers may verify your identity, maintain system integrity, and provide multiple digital IDs for users who sign documents in different roles or certification methods.

To find a digital ID file:

1. Navigate to the Add Digital ID dialog as described in “Finding an Existing Digital ID in a PKCS#12 File” on page 19.
2. Choose Browse for an existing digital ID file (Figure 3).
3. Choose Next.
4. Choose Browse and browse to the digital ID file. PKCS#12 files may reside on a network or in some local location such as C:\Users\<username>\Application Data\Adobe\<application name>\<version>\Security\ (Windows).
5. Enter the ID password if one is required.
6. Review the digital ID list and choose Finish.

2.3.2 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 10).
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.

2.3.3 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 10).
4. Choose **Change Password**.

5. Enter the old password.

6. Enter a new password and confirm it.

7. Choose **OK**.

![Figure 11 Digital ID files: Password configuration](image)

### 2.3.4 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 file 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 10).


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**.
2.3.5 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 10).
   
   **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 10).
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**.

2.3.6 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 operate in FIPS mode.
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 are your own CA.

To create a self-signed digital ID:

1. Navigate to the Add Digital ID dialog as described in “Finding an Existing Digital ID in a PKCS#12 File” on page 19.

2. Choose Create a Self-Signed Digital ID for use with Acrobat (Figure 3).

3. Choose Next.

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

- **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.
- **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**.
2.3.7 Deleting a PKCS#12 Digital ID

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

**Caution:** Because deleting an ID deletes its private key, operations that require that key will no longer be possible. Moreover, deleting the last, self-signed PKCS#12 ID (in a .pfx or .p12 file) actually deletes the ID rather than application's knowledge of the ID. 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 6**).
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.
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.

2.4.1 Finding a Digital ID in a Windows Certificate Store File

If a required digital ID file does not appear in the digital ID list, search for it and add it. You can browse to PKCS#12 files (.pfx or .p12) and Windows Certificate Store compatible files (.cer and .p7b).

For details, see “Finding an Existing Digital ID in a PKCS#12 File” on page 19.

2.4.2 Deleting a Windows Digital ID

IDs that have been added to the Windows certificate store cannot be deleted directly from the Security Settings Console. They must be removed from the Windows store by using an application such as Internet Explorer.

2.5 Managing Roaming ID Accounts and IDs

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.

Roaming IDs enable remote ID access as well as Web-based user self-registration and ID issuance from a roaming ID server and central ID management. When IDs expire, new ones can be issued and placed on a server rather than being distributed to each individual. Deployment and management therefore occurs in one location rather than on numerous client machines.
Depending on how the system is configured, users identify themselves (authenticate) to the server either with a username and password, Windows kerberos single-sign-on, or by some 3rd party method such as ArcotID.

**Note:** Roaming IDs are only used for signing and cannot be used for certificate encryption. They are subject to the same operations as described in “Generic ID Operations” on page 15

### 2.5.1 Adding a Roaming ID Account to Get a Roaming ID

Roaming IDs are only available for those with roaming ID accounts on a roaming ID server. For connection details, contact your system administrator. Once you log in to your account, the IDs associated with that account will be automatically downloaded.

To install the roaming IDs certificate:

1. Verify you have an Internet connection.

   **Tip:** If a roaming ID administrator has sent you an FDF file with the account settings preconfigured, follow the steps described in “Importing Roaming ID Account Settings” on page 154 rather than the steps below.

2. Do one of the following:
   - Navigate to the Add Digital ID dialog as described in “Finding an Existing Digital ID in a PKCS#12 File” on page 19.
   - Choose Advanced (Acrobat) or Document (Reader) > Security Settings. Then expand the left-hand tree to Roaming ID Accounts and choose Add Account from the top menu (Figure 18).

3. Choose Configure a roaming ID for use on this computer (Figure 3).

4. Choose Next.

5. In the Add a Roaming ID dialog, enter a server name and URL. In an ideal scenario, the server administrator will supply an FDF file so that you can automatically import this information.

6. Choose Next.

7. Enter your user name and password for this roaming ID server account.

8. Enter a server name and URL.

9. Choose Next.

10. Your certificate(s) will be automatically downloaded. Review the digital ID list and choose Finish.

### 2.5.2 Logging in to a Roaming ID Account

A roaming ID account is a user account on a roaming ID server containing one or more digital IDs. The login feature provides access to the IDs associated with the account. Depending on how the server administrator has set up the server, once you log in you may not be asked to supply a password again when you use that ID to sign.

To log in to a device:

1. Choose Advanced (Acrobat) or Document (Reader) > Security Settings.
2. Expand the left-hand tree to **Roaming ID Accounts** *(Figure 18)*.

3. Select an account in the right-hand panel.

4. Choose **Login**.

5. Follow the instructions in the dialogs. The workflow is vendor-based and will vary by vendor and authentication type.

**Figure 18  Roaming ID Security Settings menu items**

---

2.6 **Managing IDs Accessible via PKCS#11 Devices**

Smart cards, hardware tokens, and other PKCS#11-compliant 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.

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.

2.6.1 **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 an 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 17)*.

3. Highlight **PKCS#11 Modules and Tokens**.
4. Choose **Add Module**.

5. Browse to the device driver. On Windows, this could likely be `C:\Windows\system32\<some dll>.dll`.

6. Choose **Open**.
   The module and its IDs are automatically added to the list in the right-hand panel.

### 2.6.2 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.

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**.
2.6.3 Logging in to a Device

The digital ID login feature provides access to the IDs on a particular device or smart card.

PKCS#11 workflows are vendor-based. Whether or not additional passwords or PINs are required in a particular workflow depends on the device vendor. The login interface may be provided by the Adobe application or by the ID vendor.

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**.
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, you can set its trust level so that you can validate their signature.

Understanding what a trusted identity is and how trust levels are specified 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 31
- "Using Directory Servers to Add Trusted Identities" on page 40
- "Adding Someone to Your Trusted Identity List" on page 33
- "Managing Contacts" on page 43

### 3.1 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:

- **Extract the data from an FDF file.** Double-clicking on an emailed file or a file on some accessible directory 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 or send server details in an FDF file.

- **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 23 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
3.2 Adding Someone to Your Trusted Identity List

As shown in Figure 23, 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.

3.2.1 Adding a Certificate From a Signature

When you receive a signed document from someone whose certificate is not in your trusted identity list OR does not chain up to a trust anchor (another certificate that is trusted), the signing certificate’s validity is unknown and a question mark appears in the signature status icon. To validate the signature, you will need to trust one of the certificates in the certification chain. You could trust the signer (the end-entity certificate), one of the EE certificate issuer (an intermediate certificate), or the topmost certificate authority (the root).

Because revocation checking does not occur for certificates that are directly trusted (a trust anchor), it is best practice to trust a certificate other than the signer’s. That is, trust a certificate as high up in the chain as is practical for your signing workflows.

To add a certificate to your trusted identities list directly from a signature:

1. Right click on the signature and choose Show Signature Properties.
2. Choose Show Certificate.
3. When the Certificate Viewer appears, choose the Trust tab.
4. Choose **Add to Trusted Identities**.

5. Set the certificate trust settings as described in “Setting Certificate Trust” on page 37.

### 3.2.2 Requesting a Digital ID via Email

Email requests for digital ID information use FDF files. For details, see “Requesting a Certificate via Email” on page 163.

### 3.2.3 Importing a Certificate From a File

Adobe application's often export certificates to an FDF file. To import certificates in an FDF file, follow the instructions described in:

- “Importing Someone’s Certificate” on page 148
- “Importing Multiple Certificates” 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.

### 3.2.4 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 40.

**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 the Windows Certificate Store**

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. Choose **Import**.

9. Choose **OK** when the confirmation dialog appears.

**Figure 26  Searching for a document recipients**

3.3 Certificate Trust Settings

Contacts in the trusted identities list should be associated with one or more certificates. Certificate trust settings may be individually configured to for approval and certification signatures. Choosing to not trust a certificate does not prevent a document from displaying, but it will result in signatures having an unknown status and appearing with a question mark (**Figure 27**). For each contact, 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.
- Dynamic content and JavaScript can be configured to run on a per-certificate basis. These settings interact with application environment settings.

**Figure 28** Certificate trust settings

3.3.1 Setting Certificate Trust

Signers use their digital ID certificate to sign documents. In order for you to verify the validity of a signature, you must have explicitly trusted their certificate for signing or that certificate must chain up to another certificate you have trusted (a trusted anchor).

To trust a certificate for signing and certifying:

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**.
2. On the Trust tab, select the trust options.

**Note:** In enterprise settings, the administrator should tell you which trust settings are appropriate.

![Figure 29 Certificate trust settings](image)

- **Signatures and as a trusted root:** Trusts the certificate as 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 certificates.

  **Tip:** There is no need to make end entity certificates trusted roots if they chain up to a trust anchor. It is best practice to trust the topmost certificate that is logically 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 ICA certificate. If that certificate chains up to VeriSign, you would not want to make VeriSign a trusted root unless you wanted to trust every certificate that chains up to VeriSign.

- **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 132.

- **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 High Privilege JavaScript Options” on page 135.

  **Note:** During import, recipients of the distributed trust anchor may be able to inherit these trust settings as well as any other trust settings of certificates higher up in the chain.

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.

Policy restrictions are typically used in enterprise settings when configuring trust anchors. A restriction provides criteria the certificate chain must meet before a certificate can be used to create a valid signature. For example, a VeriSign certificate may be set as a trusted root, but a company may wish to only trust their own intermediate certificate (ICA) rather than any certificate that chains up to VeriSign. The company can issue an ICA with a certificate policy extension. By including that ICA in the certificate
chain between all end entity certificates and VeriSign and requiring the presence of that extension in Acrobat, only signers with that ICA in their certificate chain will be trusted.

To configure the Policy Restrictions tab:

- **Certificate Policies**: Enter the policy OID.
- **Description**: Enter a meaningful description.

![Figure 30 Policy restrictions](image)

4. Choose **OK** twice.

5. Choose **Close**.

### 3.3.2 Setting Certificate Policy Restrictions

Policy restrictions are typically used in enterprise settings when configuring trust anchors. A restriction provides criteria the certificate chain must meet before a signing certificate can be used to create a valid signature. For example, a VeriSign certificate may be set as a trusted root, but a company may wish to only trust their own intermediate certificates (ICAs) that chain to VeriSign rather than all certificates that chain up to VeriSign. The company can issue an ICA with a certificate policy extension. By including that ICA in the certificate chain between all end entity certificates and VeriSign and requiring the presence of that extension, only company signers will be trusted.

Policies are represented by numbers called *object identifiers* (OIDs). OIDs are usually provided by your system administrator.

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Manage Trusted Identities**.

2. Choose **Certificates from the Display drop-down list**.

3. Highlight a certificate and choose **Edit Trust**.

4. Choose the Policy Restrictions tab and enter the restrictions:
   - **Certificate Policies**: Required. Enter the policy OID.
   - **Description**: Optional. Enter a meaningful description.
3.3.3 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.

3.4 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:
  - 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, users in enterprise environments 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, sends the server configuration details in an FDF file, or tells the user how to configure the server manually.
3.4.1 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 will supply an FDF file so that you can automatically import this information.

To configure an identity directory:

1. Choose **Advanced** (Acrobat) or **Document** (Reader) > **Security Settings**.
2. Select **Directory Servers** in the left-hand list (Figure 32).
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 up to 10.
5. Choose **OK**.
3.4.2 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 32**).
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 41.
6. Choose **OK**.

3.4.3 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 32**).
3. Select a directory server from the right-hand panel.
4. Choose **Remove**.
5. When a confirmation dialog appears, choose **OK**.
3.4.4 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 32).
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.

![Digital ID Directory servers: Setting defaults](image)

3.4.5 Importing and Exporting Directory Server Settings

For details, refer to the following:

- “Importing Directory Server Settings” on page 151
- “Emailing Server Details” on page 164
- “Exporting Server Details” on page 166

3.5 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.

3.5.1 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.
3. Choose **Details**.

4. Edit the details.

5. Choose **OK**.

### 3.5.2 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 161.

### 3.5.3 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 162.
3.5.4 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 35).
3. Choose Details.
4. Choose Associate Certificate (Figure 36).
5. Select a certificate from the list.
6. Choose OK.
7. Choose OK.

3.5.5 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 35).
3. Choose Details.
4. Choose a certificate from the list.
5. Choose Remove Association (Figure 38).
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.

3.5.6 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 35).

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.
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 36**).
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 document recipients get it as well as control what they can and cannot do with a document.

For more information, refer to the following:

- “Setting up the Signing Environment” on page 48
- “Working with Signature Fields” on page 58
- “Authoring Signable Forms” on page 63
- “Checking Document Integrity” on page 68

**Best Practices for Signed Documents that will Change**

Certain workflows require that someone enter form data, provide annotations and comments, or sign a document. When the form author has authorized such changes, the application will not flag these permitted changes as warnings. 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.

### 4.1 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 authoring and signing.
4.1.1 Setting Signing Preferences

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 your preferences as described in the following sections:
   - “Requiring Preview Mode” on page 49
   - “Changing the Default Signing Method” on page 50
   - “Embedding Signature Revocation Status” on page 50
   - “Allowing Signing Reasons” on page 51
   - “Showing Location and Contact Details” on page 51
   - “Enabling Document Warning Review” on page 51
   - “Requiring Document Warning Review Prior to Signing” on page 51
   - “Enabling a Warnings Comment or Legal Attestation” on page 52

### 4.1.1.1 Requiring Preview Mode

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 allowed under PDF/SigQ-1B that may prevent a signer from knowing what they are signing. For details about PDF/SigQ, see “PDF/SigQ” on page 68.
- It informs the user the document’s level of PDF/SigQ conformance and generates a report.

1. Set View documents in preview document mode when signing.
2. Choose **Advanced Preferences**.

3. Choose the Creation tab (Figure 41).

![Figure 40 Preview document mode preference](image)

![Figure 41 Signature creation preferences](image)

### 4.1.1.2 Changing the Default Signing Method

*Continued from the previous section.*

4. *(Optional)* **Default Signing Method**: Do not change the signing method unless instructed to do so by your administrator. To learn more about signing methods, see “Changing the Default Signing Method” on page 57.

### 4.1.1.3 Embedding Signature Revocation Status

*Continued from the previous section.*

5. *(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 certificate
revocation list folder. If it is not there, the application goes online to complete the check.

### 4.1.1.4 Allowing Signing Reasons

*Continued from the previous section.*

6. *(Optional)* Set **Show reasons when signing**. 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 review this
document” or create a new one.

### 4.1.1.5 Showing Location and Contact Details

*Continued from the previous section.*

7. *(Optional)* Set **Show location and contact information when signing**. When this option is turned on,
the **Location** and **Contact Info** fields appear in the signing dialog.

### 4.1.1.6 Enabling Document Warning Review

*Continued from the previous section.*

**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 not
adversely impacted by malicious content.

8. *(Optional)* Set **Enable Reviewing of Document Warnings**: Enabling document warning review allows
signers to check document integrity prior to signing. The document can be analyzed by the PDF/SigQ
Conformance Checker 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. 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.

### 4.1.1.7 Requiring Document Warning Review Prior to Signing

*Continued from the previous section.*

**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 not adversely impacted by malicious content.

9. (Optional) 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.

### 4.1.1.8 Enabling a Warnings Comment or Legal Attestation

*Continued from the previous section.*

For certified documents that contain multimedia, comments, or other dynamic content, it 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 interactive,” or create a custom comment.

To enable warnings comments on certified documents:

10. Set Enable Reviewing of Document Warnings to When certifying a document or Always as described in “Enabling Document Warning Review” on page 51.

11. Choose OK and exit the preferences dialog.

### 4.1.2 Custom 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 42 Custom signature appearance*

![Custom signature appearance image]

**Tip:** To learn how to control appearances programmatically, developers should refer to Digital Signature Appearances.
4.1.2.1 Creating a Custom Signature

Acrobat creates a default signature from the signer’s name. However, a signature can be any graphic such as scanned signatures, text, or a combination of the two (Figure 42). 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.

4.1.2.2 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%.
3. Save the file to a PDF file.
4. Open the PDF file in Acrobat.
5. Crop the page and remove white space.
   Note: The method varies across product versions. For 8.x, choose Document > Crop Page and check Remove White Margins.
6. Save the file as SignatureLogo.pdf in C:\Documents and Settings\<user>\Application Data\Adobe\Acrobat\<version>\Security.

4.1.2.3 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 43).

Note: If you have created a watermark file as described in “Creating a Custom Watermark or Background” on page 53, 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**.

![Signature appearance: New button](image)

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**.
4.1.2.4 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.

4.1.3 Signing with a Timestamp

Timestamp Basics

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 45).
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 45 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 with a question mark icon (Figure 46).
- Trusted timestamps that have been added to the Trusted Identities list and have been explicitly trusted for signing appear with a clock icon (Figure 47).

Figure 46 Timestamps: Untrusted stamp

Figure 47 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.

Setting up a Timestamp Server

In order for a signature to use a timestamp, 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 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 150.
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 48 Timestamps: Entering server details**

4.1.4 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. Install the third party plugin. For details, use the plugin's documentation or contact your administrator.

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

3. Choose **Security** in the left-hand category list.

4. Choose **Advanced Preferences**.

5. Choose the Creation tab (**Figure 41**).
6. Select a signing method from the Default Method to Use When Signing and Encrypting Documents drop-down list.

4.2 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

4.2.1 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 Advanced > Sign & Certify > Place Signature.

2. If a confirmation dialog appears, choose OK.

3. Click and drag where the field should appear. The Digital Signature Properties dialog appears.
4. 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 49 Signature field: General properties**

![Digital Signature Properties window](image)

4.2.2 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 **Tools > Forms > Digital Signature Tool** 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 65.

4. Choose **Close**.

4.2.3 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 **Tools > Forms > Digital Signature Tool** and then double click on them OR right click and choose **Properties**.

2. Display the Appearance tab.

3. Configure the appearance options.
4. Choose Close.

Figure 52  Signature field: Appearance properties

4.2.4 Changing the Default Field Appearance

The default appearance of blank signature fields is an invisible 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.

4.2.5 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 Tools > Forms > Digital Signature Tool.
2. Right click on the field.
3. Choose Edit > <Cut, Copy, Paste, or Delete> (Figure 53).
4.2.6 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 **Tools > Forms > Digital Signature Tool**.
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 53).

4.2.7 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 **Tools > Forms > Digital Signature Tool**.
2. Right click on the field.
3. Choose **Create Multiple Copies** (Figure 53).
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.
4.3 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 63
- “Locking Fields Automatically After Signing” on page 64
- “Making a Field a Required Part of a Workflow” on page 65
- “Specifying a Post-Signing Action” on page 65
- “Unlocking a Field Locked by a Signature” on page 67

4.3.1 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 60.

Names and tooltips: Intuitive field names and tooltips facilitate authoring and signing. Unique and intuitive names 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. Names and tooltips also help signers find fields and understand how to use the form. For details, see “Specifying General Field Properties” on page 59.

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 64.

4.3.2 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 field.

   **Note:** For existing fields, place them field in edit mode by selecting Tools > Forms > Digital Signature Tool and then double click on them OR right click and choose Properties.

2. Display the Signed tab.

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

   When a field with field locking specified 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 not only validates the bytes of the PDF file, but it also compares the object
hash in the signature to the object hash from the objects in memory. This allows the application to detect attempted, but 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.

### 4.3.3 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 **Tools > Forms > Digital Signature Tool**.
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.

![Required field not signed alert](image)

### 4.3.4 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 them in edit mode by selecting **Tools > Forms > Digital Signature Tool** and then double click on them OR right click and choose **Properties**.

2. Display the Action tab.

   **Figure 57 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**

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute a Menu Item</td>
<td>Executes a specified menu command as the action.</td>
</tr>
<tr>
<td>Go to 3D View</td>
<td>Changed the view to the 3D view specified by the form author.</td>
</tr>
<tr>
<td>Go to a Page View</td>
<td>Jumps to the specified destination in the current document or in another document.</td>
</tr>
<tr>
<td>Import Form Data</td>
<td>Brings in form data from another file, and places it in the active form.</td>
</tr>
<tr>
<td>Open a File</td>
<td>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.)</td>
</tr>
<tr>
<td>Open a Web Link</td>
<td>Jumps to the specified destination on the Internet. You can use http, ftp, and mailto protocols to define your link.</td>
</tr>
<tr>
<td>Play a Sound</td>
<td>Plays the specified sound file. The sound is embedded into the PDF document in a cross-platform format that plays in Windows and Macintosh.</td>
</tr>
<tr>
<td>Play Media (Acrobat 5 Compatible)</td>
<td>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.)</td>
</tr>
<tr>
<td>Play Media (Acrobat 6 Compatible)</td>
<td>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.)</td>
</tr>
<tr>
<td>Read an Article</td>
<td>Follows an article thread in the active document or in another PDF document.</td>
</tr>
<tr>
<td>Reset a Form</td>
<td>Clears previously entered data in a form. You can control the fields that are reset with the Select Fields dialog box.</td>
</tr>
<tr>
<td>Run a JavaScript</td>
<td>Runs the specified JavaScript.</td>
</tr>
<tr>
<td>Set Layer Visibility</td>
<td>Determines which layer settings are active. Before you add this action, specify the appropriate layer settings.</td>
</tr>
<tr>
<td>Show/Hide a Field</td>
<td>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.</td>
</tr>
<tr>
<td>Submit a Form</td>
<td>Sends the form data to the specified URL.</td>
</tr>
</tbody>
</table>

### 4.3.5 Unlocking a Field Locked by a Signature

Only document authors can unlock a signature field once it has been locked. 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.

To unlock a field:

1. Press **Esc** to get Acrobat out of edit mode.
2. Right click on the signature field.
3. Choose **Clear Signature Field**.

Other fields can now be edited as usual.
4.4 Checking Document Integrity

The practices described below provide measures which increase the level of security and reliability of signing workflows by providing document authors, signers, and signature validators with a method for obtaining consistent rendering of signed documents. At a high level, checking document integrity involves the following:

- Understanding what PDF/SigQ is and whether it is relevant to your workflow.
- Knowing what preview mode does and how to use it.
- If preview mode is used, analyzing the result to determine if a document should be trusted.

4.4.1 PDF/SigQ

PDF/SigQ defines a PDF 1.7 variant that can be used to produce a document that has a deterministic and repeatable visual rendering that is consistent with the state of the document as signed. PDF/SigQ 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 compliance with PDF/SigQ is determined by the presence or absence of certain content:

- **Content with external dependencies.** These documents are **not compliant**. This type of content is not allowed in PDF/SigQ conforming documents. Examples of external content are externally referenced images or multimedia content outside of the PDF file, and TrueType fonts.

- **Interactive content.** These documents are **PDF/SigQ 1-B compliant**. Interactive content that can be suppressed by the viewing mode is allowed in PDF/SigQ 1-B conforming documents but not Level A. Interactive content which cannot be suppressed is not allowed in PDF/SigQ conforming documents.

- **No content with external dependencies or interactive content.** These documents are **PDF/SigQ 1-A compliant**.

4.4.2 Preview Mode and PDF/SigQ Conformance

Preview mode invokes the PDF/SigQ Conformance Checker which analyzes a document for compliance with the PDF/SigQ. The checker analyzes documents for content that may alter the document’s appearance and the preview mode suppresses that content if possible, thereby allowing you to view, sign, and validate the document in a static and secure state.

While conformance is not required by any government organization or standard, conformance to PDF/SigQ-1A or 1B 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 a non-compliant document and whether to trust signatures in a non-compliant document. Preview mode is recommended for both signing and validation.

The PDF/SigQ Conformance Checker can be invoked in several ways:

- **Before signing:** Choose **Advanced > Sign & Certify > Preview Document**.
- **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 Document Integrity Properties**.
4.4.3 PDF/SigQ Checker Results

Application behavior varies slightly depending on how the PDF/SigQ Conformance Checker was invoked as well as the level of document compliance with PDF/SigQ. In general however, PDF/SigQ compliance can be determined by the document message bar (DMB) text or through the View Report button on the DMB.

As shown in the following tables, PDF/SigQ conformance errors categorize content as one of the following:

- **Interactive 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.

Use the PDF/SigQ level column in Table 7 through Table 10 to determine the compliance level of the content generating the error. It may be prudent to not sign a document that is neither PDF/SigQ-1A or 1B compliant. For documents that are PDF/SigQ-1B compliant, signers may wish to analyze the document for non-conforming content in order to evaluate whether they should continue signing.

### Table 7 Interactive feature warnings

<table>
<thead>
<tr>
<th>String</th>
<th>Code</th>
<th>Description</th>
<th>PDF/ SigQ level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document contains hidden behavior</td>
<td>1000</td>
<td>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.</td>
<td>B</td>
</tr>
<tr>
<td>Comment or form field may silently change</td>
<td>1001</td>
<td>The document contains non-signature form fields. Such fields' visual appearances may change based on external variables.</td>
<td>B</td>
</tr>
<tr>
<td>Comment or form field may silently change</td>
<td>1002</td>
<td>The document contains comments. Comments' visual appearances may change based on external variables</td>
<td>B</td>
</tr>
<tr>
<td>Document may silently launch menu items</td>
<td>1003</td>
<td>The document contains named actions that may launch menu items without the user's knowledge.</td>
<td>B</td>
</tr>
<tr>
<td>Presentation elements may change appearance</td>
<td>1004</td>
<td>Presentations are not allowed since a presentation may contain animations or other elements that may change document appearance or behavior.</td>
<td>B</td>
</tr>
<tr>
<td>The document contains a dynamic form</td>
<td>1005</td>
<td>XFA-based (dynamic forms) documents are not allowed since such forms could alter the document's appearance or behavior.</td>
<td>none</td>
</tr>
<tr>
<td>Document contains links to external PDFs</td>
<td>1006</td>
<td>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.</td>
<td>B</td>
</tr>
<tr>
<td>Comment or form field may silently change</td>
<td>1007</td>
<td>Disallowed annot type: &lt;annot type&gt;. One or more form fields are associated with a 3D object, file attachment, multimedia, or other dynamic objects.</td>
<td>B</td>
</tr>
<tr>
<td>Document contains hidden behavior</td>
<td>1008</td>
<td>Disallowed action type: &lt;action type&gt;. 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.</td>
<td>B</td>
</tr>
<tr>
<td>Document contains hidden behavior</td>
<td>1009</td>
<td>The document's content is divided into layers that can be silently displayed or hidden on the fly.</td>
<td>B</td>
</tr>
</tbody>
</table>
### Table 8  PDF Content with variable rendering

<table>
<thead>
<tr>
<th>String</th>
<th>Code</th>
<th>Description</th>
<th>PDF/Sig Q level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page content may silently change</td>
<td>2004</td>
<td>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.</td>
<td>none</td>
</tr>
<tr>
<td>Document may not open in the future</td>
<td>2006</td>
<td>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.</td>
<td>none</td>
</tr>
<tr>
<td>Page content may silently change</td>
<td>2007</td>
<td>The document author has enabled image interpolation. No image interpolation is allowed.</td>
<td>none</td>
</tr>
<tr>
<td>Page content may silently change</td>
<td>2009</td>
<td>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</td>
<td>none</td>
</tr>
<tr>
<td>Page content may silently change</td>
<td>2010</td>
<td>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</td>
<td>none</td>
</tr>
<tr>
<td>Page content may silently change</td>
<td>2011</td>
<td>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</td>
<td>B</td>
</tr>
<tr>
<td>Page content may silently change</td>
<td>2012</td>
<td>Image XObject must not contain an alternate version</td>
<td>B</td>
</tr>
<tr>
<td>Text appearance may silently change</td>
<td>2013</td>
<td>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.</td>
<td>none</td>
</tr>
<tr>
<td>Text appearance may silently change</td>
<td>2014</td>
<td>Disallowed font type: &lt;font type&gt;. 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.</td>
<td>none</td>
</tr>
</tbody>
</table>

### Table 9  External Content

<table>
<thead>
<tr>
<th>String</th>
<th>Code</th>
<th>Description</th>
<th>PDF/Sig Q level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document links to external content</td>
<td>3000</td>
<td>Document links to images not in the PDF. No external XObjects allowed.</td>
<td>none</td>
</tr>
<tr>
<td>Document links to external content</td>
<td>3001</td>
<td>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.</td>
<td>B</td>
</tr>
<tr>
<td>Document links to external content</td>
<td>3002</td>
<td>Document contains external streams. The author has flagged some PDF bytes as a stream which may get data from an external source.</td>
<td>no</td>
</tr>
<tr>
<td>Document links to external content</td>
<td>3003</td>
<td>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.</td>
<td>B</td>
</tr>
</tbody>
</table>
4.4.3.1 PDF/SigQ Level A

PDF/SigQ Level A does not allow interactive content or any content with external dependencies. These documents are the most safe 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.

PDF/SigQ Level A compliance is indicated by the text on the DMB and/or the PDF/SigQ Conformance Report dialog available through the DMB’s View Report button.

\[\text{Figure 58 Document Message Bar: Level A compliance}\]

<table>
<thead>
<tr>
<th>String</th>
<th>Code</th>
<th>Description</th>
<th>PDF/SigQ level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unrecognized PDF content</td>
<td>4000</td>
<td>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 (PDF 1.8 or above).</td>
<td>none</td>
</tr>
<tr>
<td>Page content may silently change</td>
<td>4001</td>
<td>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.</td>
<td>none</td>
</tr>
<tr>
<td>PDF content contains errors</td>
<td>4002</td>
<td>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.</td>
<td>none</td>
</tr>
</tbody>
</table>

Table 10 Uncategorized warnings

4.4.3.2 PDF/SigQ Level B

PDF/SigQ 1 Level B allows documents to contain certain types of interactive content that it can suppress in preview mode. When documents are signed in preview mode, that view of the document (with the interactive content suppressed) is saved so that signature validators can use the View Signed Version feature to validate the signature.

PDF/SigQ Level B compliance is indicated by the text on the DMB and/or the PDF/SigQ Conformance Report dialog available through the DMB’s View Report button. View Report also lists discovered rich content. Suppressed content is associated with a green check (Figure 60). 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/SigQ Conformance Report dialog for information about content that prevents compliance with level A.

\[\text{Figure 59 Document Message Bar: Level B compliance}\]
4.4.3.3 No Compliance with PDF/SigQ

No compliance with PDF/SigQ means that the document contains content or behaviors which are invisible to the signer or signature validator which cannot be suppressed in preview mode.

The PDF/SigQ Conformance Report dialog automatically appears. Content which cannot be suppressed is associated with a red X. Users can highlight an error to see more detail (Figure 61). 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 60 PDF/SigQ Conformance Report: Level B compliance

Figure 61 PDF/SigQ Conformance Report: No compliance with level A or B
5

Signing with a Digital Signature

Like a conventional, handwritten signatures, 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 74
- “Signing with an Approval Signature” on page 81

5.1 Signing Basics

5.1.1 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 48.
- **Obtain a digital ID**: Get a digital ID from 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.

5.1.2 Signature Types

A document can 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 66). Use certification signatures for the following:
  - When you 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.
- **Approval Signature**: An approval signature is any signature that was applied without using the certification workflow. Any signature other 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.

5.1.3 Signing User Interface

Signing features are accessible in several ways which vary depending on whether a document already contains signature fields or signatures (Figure 62):

- **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.

![Figure 62 Digital signatures user interface](image)

5.2 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 this feature preference 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 revokes the document’s certification.

**Document Legal Defensibility**

Acrobat has a notion of a documents’ legal 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).

![Certified document indicators](image)

**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/SigQ Conformance Checker. 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 48.

**5.2.1 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 legally 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 runs the PDF/SigQ Conformance Checker 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).

### 5.2.2 Setting up a Document for Certification

#### Setting up Form Fields

When a certified document contains more than one form field, field properties such as locking, placement, 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 63.

#### Configure Document Behavior

You can customize the way a certified document behaves for recipients by using one or more of Acrobat’s advanced features. For example, you can preconfigure custom signing reasons or customize revocation checking. For details, see Chapter 7, “Controlling Signing with Seed Values”.

- Administrators can modify the registry. `bShowWarningForChanges` determines whether or not to show a warning icon on validated signature(s) if the document changes after it was signed. If false (0), the warning triangle does not appear and the status icon remains a green check and pen even if a document modified. The setting enables workflows where documents can be changed or signed multiple times without impacting the validity of previous signatures.

#### Certifying XFA Dynamic Forms

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 typically contain JavaScript and other behaviors that prevent certification unless these elements are manually approved by the author or form provider.

To set up 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.
5.2.3 Certifying a Document

Before continuing:

- Configure your application as described in “Setting up the Signing Environment” on page 48.
- Prepare the document for certification as described in “Setting up a Document for Certification” on page 77.

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 68).
   - Choosing a menu item:
     - **Advanced > Sign & Certify > Certify with Visible Signature**
     - **Advanced > Sign & Certify > Certify without Visible Signature**

     **Note:** Highlighting a field first results in signing that field. When a field is not preselected, choosing one of these menu items forces the signer to create a new field.

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** if the button exists. The button appears when the document is not PDF/SigQ-1A compliant. The document contains rich content that could adversely impact the integrity of the document. In general:
      - PDF/SigQ-1A compliant documents can always be signed.
      - PDF/SigQ-1B compliant documents contain rich content that can be suppressed in preview mode. These documents can be signed safely in preview mode.
      - Documents that are not PDF/SigQ compliant should not be signed by signers who are concerned that rich 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 16.
   - **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 65.

7. If the **Review** button appears on the dialog (an application setting), choose **Review**. This action runs the PDF/SigQ Conformance Checker 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 65 Certifying a document: Document integrity warnings**

8. If there are any document warnings in the PDF/SigQ Conformance 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 66  Certifying a document: Signature**

![Signature](image)

### 5.2.4 Certifying a Dynamic XFA Form

This information pertains only to LiveCycle 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 typically contain JavaScript and other behaviors that prevent certification unless these elements are manually approved by the author or form provider.

**Note:** There are situations with dynamic forms where the warning triangle indicates there have been changes when there are none. For example, when a document contains scripts that run before the document has finished opening, the application cannot detect whether the script has changed the document or not since it is still being constructed from the file's bytes on disk. In this case, a warning triangle appears but the application cannot provide details about what has changed.

To set up 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 67  Dynamic form certification setting**

![Scripting Setting](image)

### 5.2.5 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.
5.3 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.

**Tip:** Documents requiring multiple signatures or locked behavior typically use certification signatures. For details, see “Signing With a Certification Signature” on page 74.

5.3.1 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.
- If you want to create a new field on-the-fly as part of the signing process, choose a **Sign Document** menu item. However, 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 48.
- Prepare the document for certification as described in “Setting up a Document for Certification” on page 77.

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 68).
     - Highlight any signature field and choose **Advanced > Sign & Certify > Sign Document**.
   - **Sign a new field** by choosing **Advanced > Sign & Certify > Place Signature**.

   ![Signature field sign menu](image)

2. (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** if the button exists. The button appears when the document is not PDF/SigQ-1A compliant. The document contains rich content that could adversely impact the integrity of the document. In general:

- PDF/SigQ-1A compliant documents can always be signed.
- PDF/SigQ-1B compliant documents contain rich content that can be suppressed in preview mode. These documents can be signed safely in preview mode.
- Documents that are not PDF/SigQ compliant should not be signed by signers who are concerned that rich 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 69 Signing a document: Signature details](image)

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 16.
   - **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.

### 5.3.2 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**.

### 5.3.3 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 certification 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 Signature Fields**.
- In the Signatures tab, choose **Options > Clear All Signature Fields**.

To clear a single signature field:

1. Right click on a signature.

2. Choose **Clear Signature Field**.
6 Digital Signature Validation

When you receive a signed document it is advisable to validate its signature(s). 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 help you create trouble-free signature workflows and resolve 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 84
- “Setting up Your Environment for Signature Validation” on page 87
- “Validating Signatures Manually” on page 90
- “When the Status Icon is Not a Green Check . . .” on page 98
- “Troubleshooting Digital ID Certificates” on page 102
- “Viewing and Comparing Changes and Versions” on page 105

6.1 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.

6.1.1 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 was not expired or revoked at the time of signing.
- 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.

6.1.1.1 Authenticity Verification

A signer must have a digital ID that includes an X.509 certificate and the validator must add one of the certificates in its certificate chain to the trusted identities list and trust it for signing. The certificate must chain up to a trusted certificate and may be required to pass a revocation check which determines if it is valid.

Tip: You should not rely solely upon the visual inspection of signature status icons. Review the Signature Properties dialog for revocation and trust information as well as the signer’s certificate details.
6.1.1.2 Document Integrity Verification

The act of signing causes the application to create a message digest and then encrypt 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 there can be one of two results:

- Either the signature is valid but displays a warning triangle because the document has changed
- Or the signature is invalid because the changes were not allowed by the signer.

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.

6.1.2 Signature Status

By default, signatures are validated automatically when a document opens. You can confirm or change this behavior as described in “Setting Up Automatic Signature Validation” on page 87. Signature status is
represented by status icons and text both in the document, on the Signatures tab, and in the Signature
Properties dialog. After the application checks the signature's digital ID certificate status (is it valid) and
document integrity (has it changed since being signed), a signature's state can be either valid, unknown, or
invalid (Table 11):

- **Valid** signatures used a valid and trusted certificate and the document has not changed in ways
  specifically prohibited by the author.
- **Unknown** signatures are associated with certificates that cannot be validated or with a document that
  contains unknown changes or is in an unverified state.
- **Invalid** signatures either have an invalid certificate or the document has changed in ways specifically
  prohibited by the author.

### Table 11  Digital signature icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Status</th>
<th>Certificate and document status</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Certified &amp; Valid" /></td>
<td><strong>Certificate status</strong>: Valid. The signer used the certification process, and the signature was the first signature in the document. For details, see “Signing With a Certification Signature” on page 74. and <strong>Document integrity status</strong>: The document has not changed since it was signed or has only changed in ways specifically permitted by the certifier.</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Valid" /></td>
<td><strong>Certificate status</strong>: Valid. and <strong>Document integrity status</strong>: The document has not changed since it was signed or only contains changes allowed by a previous signer, if any.</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Valid: Changed view" /></td>
<td><strong>Certificate status</strong>: Valid. This icon may appear in a certified document, but it only appears on approval document signatures and not certification signatures. and <strong>Document integrity status</strong>: The document has changed since it was signed. The current view of the document is not the same as that which was signed. View the signed version to see what was signed. For details, see “Viewing and Comparing Changes and Versions” on page 105</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Unknown" /></td>
<td><strong>Certificate status</strong>: Unknown: The certificate has not been trusted (and is not untrusted), the revocation check could not complete, a chain could not be built to trust anchor, and so on. and <strong>Document integrity status</strong>: The document has changed since it was signed. The document the user is viewing is not the signed version.</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Unknown" /></td>
<td><strong>Certificate status</strong>: Unknown: The certificate or any certificate in the chain up to the issuing root certificate has not been trusted (trust must exist for one certificate in the chain), the revocation check could not complete, a chain could not be built to trust anchor, the certificate has been trusted for signing but not for certifying. and <strong>Document integrity status</strong>: The document has not changed since it was signed.</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Unknown" /></td>
<td><strong>Certificate status</strong>: Unverified. The certificate validation check has not executed or could not complete due to bad revocation information, and non-responding server, no network connection, etc. and <strong>Document integrity status</strong>: Unverified. The document integrity check has not executed or could not complete.</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Invalid" /></td>
<td><strong>Certificate status</strong>: The signer’s certificate was invalid. or <strong>Document integrity status</strong>: Illegal changes have been made to the document.</td>
<td></td>
</tr>
</tbody>
</table>
6.1.3 Running JavaScript and Dynamic Content

High privilege JavaScript and dynamic content in signed 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 “Setting Certificate Trust” on page 37.

6.2 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:

- **Setting Up Automatic Signature Validation**: If signatures should be validated automatically when a document opens, turn this option on.
- **Setting Digital Signature Validation Preferences**: 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.
- **Controlling Multimedia**: When certified documents may contain multimedia, specify whether or not it is allowed to run.
- **Setting High Privilege JavaScript Options**: Certain Adobe JavaScripts are defined as “high privilege.” If certified documents may contain high privilege JavaScript, set these preferences.

6.2.1 Setting Up Automatic Signature Validation

The only reason a user might NOT want to verify a signature is to gain a small increase in application speed when it opens a document. However, because the difference is often negligible (it depends on the number of signatures and whether a system administrator has customized revocation checking), it is advisable to always verify signatures. Verification tells the user two things:

- Whether the document is signed by someone with a valid certificate (identify verification).
- The document has not changed in a way that invalidates the signature (document integrity verification).

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**.

**Note**: Continue configuring signature validation options below.
6.2.2 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.

3. *(Optional)* 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).**

Signatures are created and validated by plugins. These options specify which plugin is used to verify a signature. 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. *(Optional)* If you have installed a non-default plugin, select the default method for verifying signatures. Use the default setting unless instructed to change it by a system administrator.

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 default timestamp server time specified in the Security Settings Console.
   - **Creation time**: The signature creation time.

7. Check or uncheck **Hide signature field validity icon when the signature is valid**: The icons are useful for rapidly determining signature status (Table 11). Icons cannot be hidden for invalid signatures.

   **Tip:** Continue configuring signature validation options below.

6.2.3 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.
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.

### 6.3 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. 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 84
- “Signature Status” on page 85

Validating a signature allows you to verify the signer’s identity and determine whether the displayed document is identical to what was signed (or 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.

### 6.3.1 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:
6.3.2 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 91. Signature validity can be determined by viewing its associated icon (Table 11). 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**.

6.3.3 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 (Figure 74), choose **OK**. This dialog does not appear if you have previously checked **Do not show this message again**.

3. If a dialog appears confirming all signatures have been validated, choose **OK** (Figure 75). This dialog does not appear if you have previously checked **Do not show this message again**.
6.3.4 Validating an Unknown Signature

If a signer's digital ID certificate has not been explicitly trusted, the signer is untrusted and the signer's signature validity will be unknown. Unknown signatures and signature status strings are associated with a question mark icon (Figure 76).

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 unknown identity to 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 76).
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 103.

4. When the Certificate Viewer appears, choose the Trust tab (**Figure 77**).

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 from VeriSign and it is trusted, then any certificates having VeriSign’s certificate as the root will also be trusted. It is a best practice to trust the topmost certificate as possible. 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.

8. When the Import Contact Settings dialog appears, configure the its trust levels. For details, see “Setting Certificate Trust” on page 37.

   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 question mark icon on the signature will not change until the signature is revalidated.

6.3.5 Validating a Signature for an Earlier Document Version

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.

When you open a document, the latest version is always displayed. However, it is often necessary to view the signed version in order to see what content was actually signed. When the viewed version is not the signed version, the signature status is either valid with a warning or unknown. Viewing the signed version
allows you to check if the signature is valid for a particular document version. For more about versioning, see “Document Integrity Verification” on page 85.

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 Show Signature Properties.
4. Choose View Signed Version. The application opens the signed version of the document.
5. Revalidate the signature if necessary.

Figure 79 Digital Signature Properties: Document Versioning panel

6.3.6 Validating Signature Timestamps

A timestamped signature could be valid whether or not a timestamp server’s certificate is valid. If you know a signature is timestamped or your workflow requires timestamps, read the following sections.

Local time vs 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 is 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 80).

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 80 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 with a question mark icon (Figure 81).
- Trusted timestamps that have been added to the Trusted Identities list and have been explicitly trusted for signing appear with a clock icon (Figure 82).

**Figure 81** Timestamps: Untrusted stamp

![Signature is timestamped but the timestamp could not be verified.](image)

**Figure 82** Timestamps: Trusted stamp

![Signature is timestamped.](image)

---

### 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:

- **That the timestamp was applied**: If a timestamp fails for some reason (the server cannot be found, the network is down, etc.), timestamping fails silently.
- **That the timestamp certificate is valid**: 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 (Figure 80): Timestamping failed and only the local time is used. Verify the signer used a timestamp.
   - Pen and question mark (Figure 81): A timestamp was used but you have not yet trusted the timestamp certificate. Proceed to the next step.
   - Clock (Figure 82): A timestamp with a trusted certificate was used.
Note: The following steps add a timestamp certificate to your list of trusted identities. Use these procedures when a timestamp is associated with a pen and question mark icon.

3. Choose **Show Certificate**.

4. When the Certificate Viewer appears, choose the Trust tab (Figure 77).

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 from VeriSign and it is trusted, then any certificates having VeriSign's certificate as the root will also be trusted. It is a best practice to trust the topmost certificate as possible. 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**.

8. When the Import Contact Settings dialog appears, configure the its trust levels. For details, see “Setting Certificate Trust” on page 37.

   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 question mark icon will change into a clock when the signature is revalidated.

### 6.3.6.1 When Timestamps Can’t be Verified...

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

- 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 chain up to a trusted root (the recommended choice) or it must be specified as a trusted root.

### 6.4 When the Status Icon is Not a Green Check...

**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 and pen icon for approval signatures or a blue ribbon icon for certification signatures. While these icons may be turned off in the signature appearance, they will always appear in the Signature Pane and in the Signature Properties dialog. Icons other than a green check and pen indicate the signatures have not been successfully validated and you should troubleshoot the problem.

A key tool for troubleshooting signatures is the 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. 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.

**Tip:** To display the Signature Properties dialog, right click on any signature in the document or the Signatures tab, and choose **Show Signature Properties**.

For details, see the following:

- “Green Check and Caution Triangle” on page 99
- “Question Mark and Caution Triangle” on page 99
- “Question Mark and Person” on page 100
- “Question Mark and Pen” on page 100
- “Red X and Pen” on page 101
6.4.1 Green Check and Caution Triangle

The warning triangle with a green check indicates that the signature is valid and that you are not viewing the signed version of the document. The signer’s certificate is valid, the document has changed since it was signed, and those changes have not been prohibited by the author. For example, a document author may sign with a certification signature to allow form fill-in or additional signatures. Since each signature is associated with a document version, a warning triangle means that you are not viewing the signed version.

This icon can only be associated with approval signatures. While it may appear in a certified document, it is never used for a certification signature.

Removing the warning triangle is as simple as viewing what was signed. To view a signature-specific document version:

1. Right click on a signature and choose View Signed Version. For details, see “Viewing a Signed Document Version” on page 106.
2. Revalidate the signature if necessary.
3. If there is no icon in the signature itself, view the signature in the signature pane to confirm there is no warning triangle.

Figure 85 Digital Signature Properties: Document Versioning panel

You can review the document changes as described in the following:

- “Viewing a List of Post-Signing Modifications” on page 106
- “Comparing Documents” on page 107

6.4.2 Question Mark and Caution Triangle

A question mark with a caution triangle means that the certificate validity is unknown and that the document has changed since it was signed. The question mark means you have not trusted the certificate, the revocation check could not complete, a chain could not be built to a trust anchor, and so on. The caution triangle tells you that you are not viewing the signed version of the document.

To troubleshoot signature status:

1. Verify the signer is in your trusted identity list.
2. Specify the certificate’s trust settings as described in “Setting Certificate Trust” on page 37.
3. 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.

4. Right click on the signature and choose **View Signed Version**. The signed version of the document opens. For details, see “Viewing a Signed Document Version” on page 106.

5. Revalidate the signature for the signed version if necessary.

You can review the document changes as described in the following:

• “Viewing a List of Post-Signing Modifications” on page 106
• “Comparing Documents” on page 107

### 6.4.3 Question Mark and Person

A question mark with a person means that the certificate validity is unknown and that the document has not changed since it was signed. You may have not trusted the certificate, the revocation check could not complete, or a chain could not be built to a trust anchor. If the signature is a certification signature, the certificate may have only been trusted for signing but not for certifying.

To troubleshoot signature status:

1. Verify the signer is in your trusted identity list.

2. Verify you have trusted the certificate for signing and (if necessary) certifying. Specify the certificate's trust settings as described in “Setting Certificate Trust” on page 37.

3. It may be that there was a problem with certificate revocation checking. 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.

### 6.4.4 Question Mark and Pen

A question mark and a pen means that certificate validity and document integrity status are unverified because both the certificate validation check and the document integrity check did not execute. The certificate could be OK but may not be validated for the following reasons:

• The document may have changed in which case the signature needs to be revalidated.

• If online revocation checking is required, it may have failed as a result of no online access or an application problem.

• The **Verify signatures when the document is opened** preference might be turned off and no attempt was made to check certificate validity.

Similarly, the document may have been changed in some unknown way. Manually validate the signature, and check document integrity to resolve or identify the problem.

To check certificate validity and document integrity:

1. Revalidate the signature as described in “Validating a Single Signature in Acrobat” on page 91.
2. Open the signed document version as described in “Viewing a Signed Document Version” on page 106.

3. Verify whether or not certificate revocation checking occurred. Open the Certificate Viewer’s Revocation tab (right click on a signature > Show Signature Properties > Show Certificate), and check the following:
   - If revocation checking occurred and Problems encountered is active, click the button to view a description of the problems.
   - If revocation checking did not occur and Check revocation is active, click the button to check revocation manually.

6.4.5 Red X and Pen

A red X means that either the signer’s certificate is invalid or that illegal changes have been made to the document. Since the signer’s certificate is embedded in the signature, if it is invalid, the signature will always be invalid. If illegal changes have been made to the document, there is no way to undo those changes without further changing the document illegally.

In both these cases, contact the sender to resolve the problem. The signer may need to get a new digital ID or make the signing workflow more secure so that the document cannot be changed after signing.

   Note: Policy restrictions on a trust anchor can result in signature invalidity. If you have set a policy restriction, determine if that is the problem and whether or not you wish to remove the restriction.

6.5 XFA 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. It is possible for a form to be designed with scripts to perform 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. If the form was certified, the certification signature would become invalid.

**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 signature. For example, the following changes are detected:

- Changes to the default value for an XFA object is stored in the XFA Template.
- XFA objects that become hidden or unhidden are in the XFA Form.
- Items that have a value and are cleared. These are reflected in the XFA form and XFA data.
- Items that are added to a list, a repeating subform, or anything else are reflected in the XFA form and XFA data.
- Changing poplist values. These are reflected in the XFA form and possibly the XFA data.
6.6 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

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

6.6.1 Displaying Someone’s Certificate in the Certificate Viewer

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 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 86). 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.
6.6.2 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 their 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 148.

2. Display the Details tab.

![Figure 87 Certificates: Verifying originator](image)

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 “Setting Certificate Trust” on page 37.

### 6.6.3 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**.
6.6.4 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 86).
5. Choose **Export**.
6. Email the certificate or save it to a file as described in “Exporting Your Certificate” on page 161.

6.7 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.
- Every time a document is signed, another version of the document is stored in the PDF as an incrementally numbered version.
- A signature is only valid for a specific document version. Signature X can be valid for version X, signature Y can be valid for version Y, and so on.

For details, see the following:

6.7.1 Viewing a Signed Document 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 often necessary to view the signed version in order to see what content was actually signed. When the viewed version is not the signed version, the signature status with either be valid (with a warning) or unknown.

To view the signed version of a document:

1. 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 that was signed. The status icons are removed from the signature appearance since the document is placed in preview mode.

2. Open the Signatures pane, and verify the signature is associated with a green check and pen icon.

3. Navigate to the signature field.

   Note: The signature status icon should not contain a warning triangle. Right click on the signature and choose Show Signature Properties for more detail about the signature.

Figure 89 Digital Signature Properties: Document Versioning panel

6.7.2 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, for the most thorough analysis of a signed document’s integrity, create and view the document modifications list.

To view a list of post-signing modifications:

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.
6.7.3 Comparing Documents

**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 finds and displays the differences in two versions of a document. Users can specify the type of differences to look for to verify that the appropriate changes have been made. Document Compare does not compare comments or other annotations in the document.

6.7.3.1 Comparing a Signed Version with the Current Version

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.

   **Tip:** This method compares the two versions page by page. For details about using additional comparing options, refer to the next section “Comparing Two Documents Manually” on page 107.

6.7.3.2 Comparing Two Documents Manually

You may wish to compare documents manually in order to compare a signed version with an unsigned version or to check how a document has changed over time prior to signing it. Manually comparing documents also provides you with additional control over the comparing methodology.

To compare two versions of a document:
1. Choose **Advanced > Compare Documents**.

2. In the Compare panel, select an open file from the drop-down list or use **Choose** to browse to the needed file.

   **Note:** If a file has been signed one or more times, a **Revision** drop-down list appears and each document revision will be listed. Select the revision version. The timestamp offers the most reliable chronological order of revision.

3. In the To panel, select an open file from the drop-down list or use **Choose** to browse to the needed file.

   ![Figure 91 Compare: Select file drop-down menu]

4. Set the options in the Type of Comparison panel:

   - **Page by page visual differences (Compare pages):** Find textual or graphic differences between two documents by comparing each page's bitmap (document A's page one with document B's page one, document A's page two with document B's page two, and so on.) Select the required level of detail. The higher the sensitivity, the slower Compare Pages runs. For details, see "Comparing Documents by Page" on page 109.

   - **Detailed analysis** (very slow): High sensitivity compares at 72 dpi.

   - **Normal analysis**: Normal sensitivity compares at 36 dpi.

   - **Coarse analysis** (faster): Low sensitivity compares at 18 dpi.

   - **Textual differences (Compare words):** To show which text has been inserted, deleted, or moved. Select Include Font Information to compare any formatting differences. For details, see "Comparing Documents by Word" on page 110.

   - Set the **Markup color** so that discovered changes will appear in the selected color.

   ![Figure 92 Compare: Type of comparison]

5. Set the options in the Choose Compare Report Type panel:

   - **Side by Side Report:** Creates a new document that displays the two documents in Continuous-Facing mode.

   - **Consolidated Report:** Adds markup where the differences occur in the current document. Placing the pointer over a markup in a consolidated report using the Hand tool displays the differences.

   - To compare text-based documents, select **Textual Differences** to appear in Side By Side Report format.

6.7.3.3 Comparing Documents 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 94).

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.
6.7.3.4 Comparing Documents by Word

Acrobat can compare document text rather than visual appearance, including words that are deleted, added, and moved. For example, if a word in document A, say *Acrobat*, is changed to *Acrobatic*, Acrobat interprets this change as if the word *Acrobat* was replaced by *Acrobatic*.

This reliance on a word level of granularity affects the way Acrobat detects moved blocks of text. Compare Words can only reliably detect moved strings of contiguous words at least four words long since this scheme breaks down when working with common words such as *the* or *this*. On the other hand, Compare Words can detect text blocks that were moved and changed as long as 92% of the text block stays the same.

Compare Words presents results using a temporary document with roughly the same layout as Compare Pages; that is, a two-page summary, followed by actual pages from both documents that differ, and so on. However, there are a few differences:

- The summary page displays the number of added, deleted, or moved words instead of the number of changed pages.
- In the actual document pages that differ, words that were deleted from document A are struck out; words that were added to document B are underlined.
- When a word's position within the document has been moved, it will be highlighted in a user-specified color. Click on any text block of highlighted text on one document to see the two page view that shows the corresponding text block in the other document.
Figure 96  Compare: Displaying changed text
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 signer 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 and Document Locking
- 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

### 7.1 Seed Value Basics

The 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. 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 14. Note that `certspec` and `timeStampspec` are objects that have multiple properties.
7.1.1 Changes Across Releases

Each Acrobat release results in support for additional seed values as shown in Table 13.

Table 13  Seed value changes across releases

<table>
<thead>
<tr>
<th>Seed value</th>
<th>First support for seed value</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
</table>
| certspec   | Specifies that certain certificates must be used for a particular signature field.  
8.0-7.x: Supports subject, issuer, and oid.  
8.x: Adds support for subjectDN, issuerDN, keyUsage, url, and urlType | X | X | X |
| filter     | The language-independent name of the security handler to be used when signing. | X | X | X |
| flags      | A set of bit flags controlling which properties are required.  
8.0: 16: legalAttestations, 32: shouldAddRevInfo, and 64: digestMethod. | X | X | X |
| legalAttestations | A list of legal attestations that the user can use when creating an MDP (certification) signature. | X | X | X |
| mdp        | Can be used to force a certification signature as well as to control permitted document changes. | 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 |
| subFilter  | An array of acceptable signature formats. | X | X | X |
| timestampspec | Specifies a timestamp server using the url and flags properties. | 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 |
| digestMethod | The algorithm used to create the message digest.  
6.0-7.x: MD5, SHA1.  
8.0: Adds support for SHA256, SHA384, SHA512, and RIPEMD160. | X | X |
| shouldAddRevInfo | Controls how the application does certificate and chain revocation checking. | X |

7.1.2 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 14  Seed value object properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>certspec</td>
<td>object</td>
<td>A seed value CertificateSpecifier Object. For details, see “Specifying Certificate Properties for Signing” on page 122.</td>
</tr>
<tr>
<td>digestMethod</td>
<td>array of strings</td>
<td>(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.</td>
</tr>
<tr>
<td>filter</td>
<td>string</td>
<td>The language-independent name of the security handler to be used when signing.</td>
</tr>
</tbody>
</table>
Table 14  Seed value object properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>flags</td>
<td>number</td>
<td>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: 1: filter 2: subFilter 4: version 8: reasons 16: legalAttestations (Acrobat 8.0) 32: shouldAddRevInfo (Acrobat 8.0) 64: digestMethod (Acrobat 8.0) 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.</td>
</tr>
<tr>
<td>legalAttestations</td>
<td>array of strings</td>
<td>(Acrobat 7.0) A list of legal attestations that the user can use when creating an MDP (certified) signature.</td>
</tr>
<tr>
<td>mdp</td>
<td>string</td>
<td>(Acrobat 7.0) The modification, detection, and prevention (MDP) setting to use when signing the field. Values include: allowNone default defaultAndComments While allowAll is a legal value, it cancels out the effect of mdp and no certification signature can be used for this field.</td>
</tr>
<tr>
<td>reasons</td>
<td>array of strings</td>
<td>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.</td>
</tr>
<tr>
<td>shouldAddRevInfo</td>
<td>boolean</td>
<td>(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.</td>
</tr>
<tr>
<td>subFilter</td>
<td>array of strings</td>
<td>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.</td>
</tr>
<tr>
<td>timeStampSpec</td>
<td>object</td>
<td>(Acrobat 7.0) A seed value timeStamp specifier object. It uses the url and flags properties to specify a time stamp server. For details, see “Specifying Timestamps for Signing” on page 119</td>
</tr>
<tr>
<td>version</td>
<td>number</td>
<td>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.</td>
</tr>
</tbody>
</table>

7.1.3 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 97 Seed values: JavaScript debugger](image)

### 7.2 Forcing a Certification Signature and Document Locking

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 98).

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 98  Seed value: Forcing a certification workflow

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 so that changes will not invalidate the 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. 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. 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 name and uses the seed value method (Example 7.1).
3. Set the mdp value:
   - **allowNone**: Do not use legalAttestations.
   - **default**: Allow form field fill-in, including signing.
   - **defaultAndComments**: Allow form field fill-in 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 and Document Locking" on page 115.

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 99 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 98).

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

**Example 7.1 Seed value: mdp**

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

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

7.3 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.

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 15**, 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 name and uses the seed value method (**Example 7.2**).
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 100**).
5. Run the JavaScript, save the document, and test the field.

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

**Example 7.2  Seed value: Custom signing reason**

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

f.signatureSetSeedValue(
    { reasons: ["This is a reason", "This is a better reason"],
            flags: 8
    } )
```
7.4 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 `timeStampspec` specifier object’s `url` and `flags` properties to specify a timestamp server.

To specify a timestamp server:

1. Create a signature field with an intuitive name and tooltip.
2. Create the JavaScript that gets the field name and uses the seed value method (Example 7.3).
3. Provide a URL for the `timeStampspec` 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 will be automatically timestamped on signing. If the application cannot find the server, an error appears (Figure 101).
   - If it is not required, the field will be 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 7.3  Timestamp server seed value**

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

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

Figure 101 Time stamp server error

7.5 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.

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 (Adbe.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 name and uses the seed value method (Example 7.4).

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 7.4  Seed value: Specifying signature components

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

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

7.6 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 17.

**Tip:** 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 name and uses the seed value method (Example 7.5).

3. Specify the digestMethod. This can be an array of comma-separated items such as ['RIPEMD160','SHA384'].
4. Run the JavaScript, save the document, and test the field.

**Example 7.5  Hash algorithm seed value**

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

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

### 7.7 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 name and uses the seed value method (**Example 7.5**).
3. Set `shouldAddRevInfo` to true.
4. Run the JavaScript, save the document, and test the field.

**Example 7.6  Hash algorithm seed value**

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

f.signatureSetSeedValue({
    shouldAddRevInfo: true
});
```

### 7.8 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
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 102 Seed value: Specifying certificates for signing**

![Diagram showing how to specify certificates for signing]

**Table 17 Seed values: certSpec properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>flags</td>
<td>number</td>
<td>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: 1: subject 2: issuer 4: oid 8: subjectDN (Acrobat 8) 16: issuerDN (Acrobat 8) 32: keyUsage (Acrobat 8) 64: url (Acrobat 8) If this field is not present, all properties are optional. Usage: 1 specifies subject, 3 specifies subject and issuer, and 6 specifies issuer and oid. If this field is not present, all properties are optional.</td>
</tr>
<tr>
<td>issuer</td>
<td>array of certificate objects</td>
<td>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 &quot;/c/test/root.cer&quot;.</td>
</tr>
</tbody>
</table>
### Table 17 Seed values: certSpec properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>keyUsage</td>
<td>array of integers</td>
<td>(Acrobat 8.0) Integers in HEX that specify the keyUsage extension that must be present in the signing certificate. Each integer is constructed as follows: There are two bits used for each keyUsage type (defined in RFC 3280) starting from the least significant bit: digitalSignature(bits 2,1) nonRepudiation(4,3) keyEncipherment(6,5) dataEncipherment(8,7) keyAgreement(10,9) keyCertSign(12,11) cRLSign(14,13) encipherOnly(16,15) decipherOnly(18,17) The value of the two bits have the following semantics: 00: The corresponding keyUsage is not allowed. 01: The corresponding keyUsage is required. 10 and 11: The state of the corresponding keyUsage doesn’t matter. For example, if it’s required that keyUsage must require digitalSignature and the state of all other’s doesn’t matter, then the corresponding integer would be 0x7FFFFFFD. That is, to represent digitalSignature, set 01 for bits 2 and 1 respectively, and set 11 for all other keyUsage types.</td>
</tr>
<tr>
<td>oid</td>
<td>array of strings</td>
<td>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 issuer property is present. oid and issuer can be used together to specify a certificate that has the selected policy.</td>
</tr>
<tr>
<td>subject</td>
<td>array of certificate objects</td>
<td>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 [&quot;/c/test/root.cer&quot;].</td>
</tr>
<tr>
<td>subjectDN</td>
<td>array of certificate objects</td>
<td>(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.</td>
</tr>
<tr>
<td>url</td>
<td>string</td>
<td>A URL that can be used to enroll for a new certificate if a matching one is not found, such as <a href="https://aardvark.corp.example.com/">https://aardvark.corp.example.com/</a>. Works in conjunction with urlType (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.</td>
</tr>
<tr>
<td>urlType</td>
<td>string</td>
<td>(Acrobat 8.0) The url type. If this attribute isn’t present, it’s assumed that the url 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.</td>
</tr>
</tbody>
</table>
7.8.1 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:

```javascript
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 .cer files in a DER format.

3. Create the JavaScript that gets the field name and uses the seed value method. Use `security.importFromFile` to get the DER-encoded certificates from their installed location (Example 7.7).

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 7.7 Certificate issuer and subject seed value**

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

var mySubjectCert = security.importFromFile("Certificate", 
"/C/Temp/nebwhifflesniet_DER.cer");
var myIssuerCert = security.importFromFile("Certificate", 
"/C/Temp/nebsCompany_DER.cer");
```
7.8.2 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 17). 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 name and uses the seed value method (Example 7.8).
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 17). For example:
      - digitalSignature is disallowed and non repudiation is required, and other values don’t matter: 111111111111110100. Convert to HEX: 3FFF4
      - digitalSignature is required and non repudiation is disallowed, and other values don’t matter: 111111111111110001. 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 0x7FFFFFF1.
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 7.8 Certificate key usage seed value

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

f.signatureSetSeedValue({
    certspec: {
        keyUsage: [0x7FFFFFF1], //Set KeyUsage to "digitalSignature"
        flags: 32 //Require keyUsage
    },
});
```
7.8.3 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 name and uses the seed value method (Example 7.9).

3. Specify the issuer.

4. Specify the oid. A policy OID is part of the value of the certificate's certificate policy field (Figure 103).

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.

Example 7.9    Certificate policy seed value

```javascript
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"],
```
7.8.4 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 name and uses the seed value method (Example 7.10).
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 7.10  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/",
   }
});
```

7.8.5 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 name and uses the seed value method (Example 7.11).
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 7.11 Roaming ID seed value**

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

f.signatureSetSeedValue{
    certspec: {
        urlType: "ASSP",
    }
}
```

7.9 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 7.12** 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 7.12 Automating signing tasks**

```javascript
//**************************************************************************
//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.addMenuItem
{
    cName: "Request Employee Signature",
    cGroupName: "Tools",
    cLabel: "Request Employee Signature",
    cHide: false
}
```
//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; //hiden 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",O:"Example Trust Services",C:"US"};
        var myusage = {endUserSigning:true};
        var ExampleRootCertBinary = "308204A130820389A00302010202043E1CBD28300D06092A864886F70D0101050303069310
        " //-----------<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
// 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");
}
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:

- “Controlling Multimedia” on page 132
- “Setting High Privilege JavaScript Options” on page 135
- “Working with Attachments” on page 136
- “Controlling Access to Content Outside a PDF” on page 142
- “Internet URL Access” on page 142

8.1 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 104 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 106).

**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 disallow dynamic content will have no effect.

**Figure 105 Multimedia behavior workflow**

8.1.1 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 (Windows): Edit > Preferences > **Multimedia Trust**
   - Acrobat (Macintosh): **Acrobat** > Preferences > **Multimedia Trust**
   - Adobe Reader (Windows): **Edit** > Preferences > **Multimedia Trust**
   - Adobe Reader (Macintosh): **Adobe Reader** > Preferences > **Multimedia Trust**
2. From the **Display Permissions for** radio buttons, choose **Trusted documents** or **Non-trusted documents**. The Trust Manager displays the selected trust preferences (Figure 106).

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**.

### 8.1.2 Controlling Multimedia in Certified Documents

**Note**: Multimedia and other dynamic content represents 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.

For details about setting certificate trust, see “Setting Certificate Trust” on page 37.

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 133.
- 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 features.

8.2 Setting High Privilege JavaScript Options

8.2.1 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 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 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.

### 8.2.2 Javascript and Certified Documents

Whether JavaScript runs in certified documents depends on whether you have explicitly trusted the sender’s digital ID certificate 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 multimedia, then JavaScript will run in that document. For example, some enterprises may issue a MyCompany certificate that allows JavaScript. If all employee certificates use MyCompany as a trust anchor, then they can send and receive certified documents within the company that contain working JavaScript.

**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.

To block or allow execution of all JavaScript from the toolbar:

1. Choose one of the following:
   - Acrobat (Windows): Edit > Preferences > JavaScript
   - Acrobat (Macintosh): Acrobat > Preferences > JavaScript
   - Adobe Reader (Windows): Edit > Preferences > JavaScript
   - Adobe Reader (Macintosh): Adobe Reader > Preferences > JavaScript
2. Check or uncheck Enable menu items JavaScript execution privileges.
3. Choose OK.
4. If you need to enable JavaScript in certified documents, set certificate trust as described in “Setting Certificate Trust” on page 37.

**Figure 107 JavaScript Security option**

---

### 8.3 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 137
- “Adding Files to the Black and White Lists” on page 140
- “Resetting the Black and White Lists” on page 141
8.3.1 Default Behavior: Black and White Lists

Exercise caution when attaching files to a PDF since some content may adversely impact document integrity or even the document’s operating environment. To mitigate the risk inherent in attachments:

- Know what the content is and from where it originated.
- Be aware of dangerous file types and how the application manages those types. Adobe applications maintain Black Lists and White Lists which control application behavior.
- Prevent attachments from opening other files and launching applications. This is the default behavior. For details about changing this behavior, see “Allowing Attachments to Launch Applications” on page 141.

Black Lists and White Lists

The Acrobat family of products always allow you to open and save PDF and FDF file attachments. However, attachments represent a potential security risk because they can contain malicious content, open other dangerous files, or launch applications. Certainly file types such as .bin, .exe, .bat, and so on will be recognized as threats by most users.

The applications store a list of some of these good (white) and bad (black) file types in the registry (Table 18). Application behavior is controlled by the file type’s membership in a list:

- **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.

![Attachment: Dangerous type warning](Figure 108)

**Why Attach a File that’s on the Black List?**

You can attach files types that are on the black list because a document recipient may have a less restrictive black list than you (the sender). While the recipient may be able to open the file, the attacher will not be able to execute or open it from within the application. Attempting to open a prohibited file type results in a warning that the action is not allowed (Figure 109).
**Figure 109 Attachment: Cannot open warning**

![Attachment: Cannot open warning](image)

**Table 18 Default prohibited file types**

<table>
<thead>
<tr>
<th>Extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.ade</td>
<td>Access Project Extension (Microsoft)</td>
</tr>
<tr>
<td>.adp</td>
<td>Access Project (Microsoft)</td>
</tr>
<tr>
<td>.app</td>
<td>Executable Application</td>
</tr>
<tr>
<td>.asp</td>
<td>Active Server Page</td>
</tr>
<tr>
<td>.bas</td>
<td>BASIC Source Code</td>
</tr>
<tr>
<td>.bat</td>
<td>Batch Processing</td>
</tr>
<tr>
<td>.bz</td>
<td>Bzip UNIX Compressed file</td>
</tr>
<tr>
<td>.bz2</td>
<td>Bzip 2 UNIX Compressed file (replaces BZ)</td>
</tr>
<tr>
<td>.cer</td>
<td>Internet Security Certificate file (MIME x-x509-ca-cert)</td>
</tr>
<tr>
<td>.chm</td>
<td>Compiled HTML Help</td>
</tr>
<tr>
<td>.class</td>
<td>Java Class file</td>
</tr>
<tr>
<td>.cmd</td>
<td>DOS CP/M Command file, Command file for Windows NT</td>
</tr>
<tr>
<td>.com</td>
<td>Command</td>
</tr>
<tr>
<td>.command</td>
<td>Mac OS Command Line executable</td>
</tr>
<tr>
<td>.cpl</td>
<td>Windows Control Panel Extension (Microsoft)</td>
</tr>
<tr>
<td>.crt</td>
<td>Certificate file</td>
</tr>
<tr>
<td>.csh</td>
<td>UNIX csh shell script</td>
</tr>
<tr>
<td>.exe</td>
<td>Executable file</td>
</tr>
<tr>
<td>.fxp</td>
<td>FoxPro Compiled Source (Microsoft)</td>
</tr>
<tr>
<td>.gz</td>
<td>Gzip Compressed Archive</td>
</tr>
<tr>
<td>.hex</td>
<td>Macintosh BinHex 2.0 file</td>
</tr>
<tr>
<td>.hlp</td>
<td>Windows Help file</td>
</tr>
<tr>
<td>.hqx</td>
<td>Macintosh BinHex 4 Compressed Archive</td>
</tr>
<tr>
<td>.hta</td>
<td>Hypertext Application</td>
</tr>
<tr>
<td>.inf</td>
<td>Information or Setup file</td>
</tr>
<tr>
<td>.ini</td>
<td>Initialization/Configuration file</td>
</tr>
<tr>
<td>.ins</td>
<td>IIS Internet Communications Settings (Microsoft)</td>
</tr>
<tr>
<td>.isp</td>
<td>IIS Internet Service Provider Settings (Microsoft)</td>
</tr>
<tr>
<td>.its</td>
<td>Internet Document Set, International Translation</td>
</tr>
<tr>
<td>Extension</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>.job</td>
<td>Windows Task Scheduler Task Object</td>
</tr>
<tr>
<td>.js</td>
<td>JavaScript Source Code</td>
</tr>
<tr>
<td>.jse</td>
<td>JScript Encoded Script file</td>
</tr>
<tr>
<td>.ksh</td>
<td>UNIX ksh shell script</td>
</tr>
<tr>
<td>.lnk</td>
<td>Windows Shortcut file</td>
</tr>
<tr>
<td>.lzh</td>
<td>Compressed archive (LH ARC)</td>
</tr>
<tr>
<td>.mad</td>
<td>Access Module Shortcut (Microsoft)</td>
</tr>
<tr>
<td>.maf</td>
<td>Access (Microsoft)</td>
</tr>
<tr>
<td>.mag</td>
<td>Access Diagram Shortcut (Microsoft)</td>
</tr>
<tr>
<td>.mam</td>
<td>Access Macro Shortcut (Microsoft)</td>
</tr>
<tr>
<td>.maq</td>
<td>Access Query Shortcut (Microsoft)</td>
</tr>
<tr>
<td>.mar</td>
<td>Access Report Shortcut (Microsoft)</td>
</tr>
<tr>
<td>.mas</td>
<td>Access Stored Procedures (Microsoft)</td>
</tr>
<tr>
<td>.mat</td>
<td>Access Table Shortcut (Microsoft)</td>
</tr>
<tr>
<td>.mau</td>
<td>Media Attachment Unit</td>
</tr>
<tr>
<td>.mav</td>
<td>Access View Shortcut (Microsoft)</td>
</tr>
<tr>
<td>.maw</td>
<td>Access Data Access Page (Microsoft)</td>
</tr>
<tr>
<td>.mda</td>
<td>Access Add-in (Microsoft), MDA Access 2 Workgroup (Microsoft)</td>
</tr>
<tr>
<td>.mde</td>
<td>Access MDE Database file (Microsoft)</td>
</tr>
<tr>
<td>.mdt</td>
<td>Access Add-in Data (Microsoft)</td>
</tr>
<tr>
<td>.mdw</td>
<td>Access Workgroup Information (Microsoft)</td>
</tr>
<tr>
<td>.mdz</td>
<td>Access Wizard Template (Microsoft)</td>
</tr>
<tr>
<td>.msc</td>
<td>Microsoft Management Console Snap-in Control file (Microsoft)</td>
</tr>
<tr>
<td>.msi</td>
<td>Windows Installer file (Microsoft)</td>
</tr>
<tr>
<td>.msp</td>
<td>Windows Installer Patch</td>
</tr>
<tr>
<td>.mst</td>
<td>Windows SDK Setup Transform Script</td>
</tr>
<tr>
<td>.ocx</td>
<td>Microsoft Object Linking and Embedding (OLE) Control Extension</td>
</tr>
<tr>
<td>.ops</td>
<td>Office Profile Settings file</td>
</tr>
<tr>
<td>.pcd</td>
<td>Visual Test (Microsoft)</td>
</tr>
<tr>
<td>.pif</td>
<td>Windows Program Information file (Microsoft)</td>
</tr>
<tr>
<td>.prf</td>
<td>Windows System file</td>
</tr>
<tr>
<td>.prg</td>
<td>Program file</td>
</tr>
<tr>
<td>.pst</td>
<td>MS Exchange Address Book file, Outlook Personal Folder file (Microsoft)</td>
</tr>
<tr>
<td>.rar</td>
<td>WinRAR Compressed Archive</td>
</tr>
<tr>
<td>.reg</td>
<td>Registration Information/Key for Windows 95/98, Registry Data file</td>
</tr>
<tr>
<td>.scf</td>
<td>Windows Explorer Command</td>
</tr>
</tbody>
</table>
8.3.2 Adding Files to the Black and White Lists

Users can indirectly manage the registry list of which file types can be opened and saved. In other words, the list in Table 18 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).

To add a file to a black or white list, attach the new file type to a document and then try to open it:

1. Choose Document > Attach a File and attach a file type not on the black or white list (e.g. myfile.xyz).
2. Open the file by highlighting it in the Attachments pane and choosing Open.
3. When the Launch Attachment dialog appears, choose one of the following (Figure 110):
   - **Open this file**: Opens the files without changing the registry list.

<table>
<thead>
<tr>
<th>Extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.scr</td>
<td>Windows Screen Saver</td>
</tr>
<tr>
<td>.sct</td>
<td>Windows Script Component, Foxpro Screen (Microsoft)</td>
</tr>
<tr>
<td>.sea</td>
<td>Self-expanding archive (used by Stuffit for Mac files and possibly by others)</td>
</tr>
<tr>
<td>.shb</td>
<td>Windows Shortcut into a Document</td>
</tr>
<tr>
<td>.shs</td>
<td>Shell Scrap Object file</td>
</tr>
<tr>
<td>.sit</td>
<td>Compressed archive of Mac files (Stuffit)</td>
</tr>
<tr>
<td>.tar</td>
<td>Tape Archive file</td>
</tr>
<tr>
<td>.tgz</td>
<td>UNIX Tar file Gzipped</td>
</tr>
<tr>
<td>.tmp</td>
<td>Temporary file or Folder</td>
</tr>
<tr>
<td>.url</td>
<td>Internet Location</td>
</tr>
<tr>
<td>.vb</td>
<td>VBScript file or Any VisualBasic Source</td>
</tr>
<tr>
<td>.vbe</td>
<td>VBScript Encoded Script file</td>
</tr>
<tr>
<td>.vbs</td>
<td>VBScript Script file, Visual Basic for Applications Script</td>
</tr>
<tr>
<td>.vsmacros</td>
<td>Visual Studio .NET Binary-based Macro Project (Microsoft)</td>
</tr>
<tr>
<td>.vss</td>
<td>Visio Stencil (Microsoft)</td>
</tr>
<tr>
<td>.vst</td>
<td>Visio Template (Microsoft)</td>
</tr>
<tr>
<td>.vsw</td>
<td>Visio Workspace file (Microsoft)</td>
</tr>
<tr>
<td>.webloc</td>
<td>Mac OS Finder Internet Location</td>
</tr>
<tr>
<td>.ws</td>
<td>Windows Script file</td>
</tr>
<tr>
<td>.wsc</td>
<td>Windows Script Component</td>
</tr>
<tr>
<td>.wsh</td>
<td>Windows Script Host Settings file</td>
</tr>
<tr>
<td>.zop</td>
<td>Compressed Archive file</td>
</tr>
<tr>
<td>.zlo</td>
<td>ZoneLabs ZoneAlarm Mailsafe Renamed .PIF file</td>
</tr>
<tr>
<td>.zoo</td>
<td>An early compressed file format</td>
</tr>
</tbody>
</table>

Table 18 Default prohibited file types
Always allow opening files of this type: Adds the file type to the white list and prevents future warnings.

Never allow opening files of this type: Adds the file type to the black list and does not open it.

4. Choose OK.

Figure 110 Launch Attachment dialog

8.3.3 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 left-hand tree.

3. Choose Restore (Figure 111).

Figure 111 Attachment panel in Trust Manager

8.3.4 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 left-hand tree.

3. Configure Allow opening of non-PDF file attachments with external applications (Figure 111):
   - Checked: Default. The application uses its stored black list to determine what that file can do. Any file on the black list will still not be allowed to open a file or launch an application.
   - Unchecked: Documents will never open other files or launch applications. Use this option if a higher level of security is needed.
8.4 Controlling Access to Content Outside a PDF

Your application can inform you when a PDF file is attempting to access external content. In the PDF world, external content is a URL or some file specification 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 Trust Manager in the left-hand tree.

3. Configure External Content:
   - **Checked**: Default.
   - **Unchecked**: The most secure behavior does not enable external streams. When external streams are enabled, PDF files that contain an embedded switch defining a host address that can silently transmit data to and from the remote host.

8.5 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.

You can control Internet access via the Manage Internet Access dialog (Figure 113). Controls are provided for the following:

- “Turning Internet Access Off and On” on page 143
- “Allowing and Blocking Specific Web Sites” on page 144
8.5.1 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 left-hand tree.
3. Choose **Change Settings** in the **Internet Access...** panel.
4. Choose **Allow all web sites** or **Block all web sites** (Figure 113).
5. Choose **OK**.
8.5.2 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 115). When you check Remember my action for this site, the site is added to your URL white or black list.

Figure 115 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 left-hand tree.
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.
Sharing Settings & Certificates with FDF

Acrobat and Adobe Reader use 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 files with these extensions are opened by the required application when opened in 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 Policy Server, LDAP directory servers, roaming credential servers, and timestamp servers.
- Creation by a user (through the application) or by a server programmatically.
- Sharing via networked directories or as email attachments.

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:

- Importing Application Settings and Digital ID Data
- “Responding to an Emailed Request for a Digital ID” on page 146
- “Importing Someone’s Certificate” on page 148
- “Importing Multiple Certificates” on page 149
- “Importing Timestamp Server Settings” on page 150
- “Importing Directory Server Settings” on page 151
- “Importing Adobe Policy Server Settings” on page 152
9.1 Importing Application Settings and Digital ID Data

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 correct workflow.

- For digital ID information, importing it into the Trusted Identity Manager.
- For server settings, importing it with the Security Settings Console.

Figure 116 FDF Email attachment

9.1.1 Responding to an Emailed 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 (apply 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 digital ID information 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 emailed digital ID request:
1. Double click the attached FDF file.

2. Choose **Email your Certificate**.

   **Figure 117 Emailing your certificate**

   ![Emailing your certificate](image)

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 118 Selecting a digital ID**

   ![Selecting a digital ID](image)

4. Choose **Select**.

5. Review the email details. You can edit the To, Subject, and Body fields.

6. Choose **Email**.

7. Send the email through your mail application.
9.1.2 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 (Figure 116) 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 120.

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 155.

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**.
9.1.3 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 (Figure 116) or may reside in a networked directory.

   **Figure 120 Importing multiple certificates**

2. If the FDF file is signed 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**.
   - If the checkbox is selected, all contacts associated with this certificate will accept 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 121 Making a contact a trusted identity

9.1.4 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. Verify the signature properties if needed (Figure 122).

   Note: If the FDF is not signed, the Signature panel will display Not signed and the Signature Properties button will be disabled.
3. Review the timestamp server list.

   **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 123 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.

**9.1.5 Importing Directory Server Settings**

In enterprise settings, servers do not usually have to be manually configured. 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 124).

   Note: If the FDF is not signed, the Signature panel will display Not signed and the Signature Properties button will be disabled.

   Figure 124 Digital ID Directory servers: Importing


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.

9.1.6 Importing Adobe Policy Server Settings

In enterprise settings, servers do not usually have to be manually configured. 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 Policy Servers in the left-hand list, and choosing Import.

2. Review the sender’s details. Verify the signature properties if needed (Figure 124).

   **Note:** If the FDF is not signed, the Signature panel will display Not signed and the Signature Properties button will be disabled.

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.

4. Choose OK.

5. Choose Import.

6. If you do not already have a default Adobe Policy 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 Policy Servers list in the Security Settings Console.
9.1.7 Importing Roaming ID Account Settings

In enterprise settings, these servers do not usually have to be manually configured. 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. Verify the signature properties if needed (Figure 127).

   **Note:** If the FDF is not signed, the Signature panel will display *Not signed* and the **Signature Properties** button will be disabled.

   **Figure 127** Importing roaming ID server settings

3. Choose **Import**.

4. Verify the roaming ID account name and server URL.

   **Figure 128** 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.
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.

---

**9.1.8 Importing a Trust Anchor and Setting Trust**

Users occasionally need to import a trust anchor into their trusted identities list 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.

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**.

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**.

5. On the Trust tab, select the trust options.

**Note:** In enterprise settings, the administrator should tell you which trust settings are appropriate.

**Figure 131 Certificate trust settings**

- **Signatures and as a trusted root**: Trusts the certificate as 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 certificates.

  **Tip:** There is no need to make end entity certificates trusted roots if they chain up to a trust anchor. It is best practice to trust the topmost certificate that is logically 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 ICA certificate. If that certificate chains up to VeriSign, you would not want to make VeriSign a trusted root unless you wanted to trust every certificate that chains up to VeriSign.

- **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 132.
- **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 High Privilege JavaScript Options” on page 135.

  **Note**: During import, recipients of the distributed trust anchor may be able to inherit these trust settings as well as any other trust settings of certificates higher up in the chain.

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.

   Policy restrictions are typically used in enterprise settings when configuring trust anchors. A restriction provides criteria the certificate chain must meet before a certificate can be used to create a valid signature. For example, a VeriSign certificate may be set as a trusted root, but a company may wish to only trust their own intermediate certificate (ICA) rather than any certificate that chains up to VeriSign. The company can issue an ICA with a certificate policy extension. By including that ICA in the certificate chain between all end entity certificates and VeriSign and requiring the presence of that extension in Acrobat, only signers with that ICA in their certificate chain will be trusted.

   To configure the Policy Restrictions tab:

   - **Certificate Policies**: Enter the policy OID.
   - **Description**: Enter a meaningful description.

   ![Figure 132 Policy restrictions](image)

7. Choose **OK** twice.

8. Choose **Close**.

### 9.2 Exporting Application Settings and Digital ID Data

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.

Figure 133 Signing an FDF file

9.2.1 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 higher up in the chain. For example, if the root certificate is trusted, then any certificates chaining up 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 should occur 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.
9.2.1.1 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.


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 135).

   **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 160

   **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.


5. Select a certificate in the chain that appears in the left-hand window.

   **Figure 134 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 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.

12. Continue with the workflow until the trusted root is emailed or placed in a directory where your intended recipients can find it.

### 9.2.1.2 Providing Instructions to the Trusted Root Recipients

For details, see “Importing a Trust Anchor and Setting Trust” on page 155.

### 9.2.1.3 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:

1. Choose **Advanced** (Acrobat) or **Document** (Adobe Reader) > **Manage Trusted Identities**.

2. Choose **Certificates** in the **Display** drop-down list.
3. Highlight the needed certificate.
4. Choose Edit Trust.
5. Display the Trust tab.
6. Set the trust level as described in “Importing a Trust Anchor and Setting Trust” on page 155.

### 9.2.2 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 161
  - “Saving Your Digital ID Certificate to a File” on page 162
- To export any certificate displayed in the Certificate Viewer, choose Export on the General tab.

#### 9.2.2.1 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 162 and then send the file as an attachment using your web-based email program.

To email a digital ID certificate:

2. Select **Digital IDs** in the left-hand tree.

3. Highlight an ID in the list on the right.

4. Choose **Export**.

5. Choose **Email the data to someone** (Figure 136).

![Figure 136 Digital ID: ID export options](image)

6. Choose **Next**.

7. Enter the recipient’s email address and any other optional information.

![Figure 137 Emailing a digital ID](image)

8. Choose **Email**.

9. When the email program opens, send the email.

### 9.2.2.2 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 136).

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**.

**9.2.3 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 138 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.

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.

9.2.4 Emailing Server Details

Adobe Policy 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:


2. Select a server category from the left-hand list.

3. Select a server from the right-hand panel.

4. Choose Export.
5. Choose *Email the exported data* to email the FDF file.

6. Choose *Next*.

The Identity panel (Figure 142) 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 124). 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 143 Digital ID Directory servers: Email details

10. Choose Next.

11. Review the export details.


9.2.5 Exporting Server Details

Adobe Policy 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:


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 141).

6. Choose Next.

   The Identity panel (Figure 142) 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 124). 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**.
# Glossary of Security Terms

## Table 81 Security Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>.apf</strong></td>
<td>See Adobe Profile Files.</td>
</tr>
<tr>
<td><strong>.cer</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>.p12</strong></td>
<td>See PKCS#12.</td>
</tr>
<tr>
<td><strong>.p7b</strong></td>
<td>See PKCS#7.</td>
</tr>
<tr>
<td><strong>.p7c</strong></td>
<td>See PKCS#7.</td>
</tr>
<tr>
<td><strong>.pfx</strong></td>
<td>See PKCS#12.</td>
</tr>
<tr>
<td><strong>Adobe Profile Files</strong></td>
<td>Adobe's legacy certificate format not used after Acrobat 5. The certificates are stored in .apf files.</td>
</tr>
<tr>
<td><strong>approval signature</strong></td>
<td>A signature used to indicate approval of, or consent on, the document terms.</td>
</tr>
<tr>
<td><strong>CA</strong></td>
<td>See certificate authority.</td>
</tr>
<tr>
<td><strong>CDS</strong></td>
<td>See Certified Document Services.</td>
</tr>
<tr>
<td><strong>CDS digital ID</strong></td>
<td>A digital ID issued by a certified document services provider.</td>
</tr>
<tr>
<td><strong>CDS digital ID certificate</strong></td>
<td>See CDS digital ID.</td>
</tr>
<tr>
<td><strong>certificate authority (CA)</strong></td>
<td>An entity that issues trusted roots.</td>
</tr>
<tr>
<td><strong>certificates</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>certification signature</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>certified document</strong></td>
<td>A document to which a certification signature has been applied.</td>
</tr>
<tr>
<td><strong>Certified Document Services (CDS)</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>certify or certifying</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>CRL</strong></td>
<td>See Certificate Revocation List.</td>
</tr>
<tr>
<td><strong>Certificate Revocation List (CRL)</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>CSP</strong></td>
<td>See Cryptographic Service Provider</td>
</tr>
<tr>
<td><strong>Cryptographic Service Provider</strong></td>
<td>Application software that allows it to use MSCAPI to communicate with cryptographic module APIs such as PKCS#11 modules, PFX files, and so on</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>digital ID</td>
<td>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.</td>
</tr>
<tr>
<td>digital signature</td>
<td>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.</td>
</tr>
<tr>
<td>digitally sign</td>
<td>To apply a digital signature using a digital ID.</td>
</tr>
<tr>
<td>EE</td>
<td>See end entity certificate.</td>
</tr>
<tr>
<td>electronic signatures</td>
<td>A digital signature.</td>
</tr>
<tr>
<td>embedded JavaScript</td>
<td>JavaScript that exists within a document rather than that which is executed from the JavaScript Console or through a batch process.</td>
</tr>
<tr>
<td>embedded validation response</td>
<td>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.</td>
</tr>
<tr>
<td>end entity certificate (EE)</td>
<td>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.</td>
</tr>
<tr>
<td>GeoTrust</td>
<td>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.</td>
</tr>
<tr>
<td>ICA</td>
<td>See intermediate certificate authority.</td>
</tr>
<tr>
<td>individual digital ID</td>
<td>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.</td>
</tr>
<tr>
<td>intermediate certificate authority (ICA)</td>
<td>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.</td>
</tr>
<tr>
<td>MSCAPI</td>
<td>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.</td>
</tr>
<tr>
<td>Online Certificate Status Protocol (OCSP)</td>
<td>OCSP defines a protocol for determining the revocation status of a digital certificate without requiring a CRL. Unlike CRL, OCSP obviates the need to frequently download updates to keep certification status lists current. Acrobat's OCSP revocation checker adheres to RFC 2560.</td>
</tr>
<tr>
<td>organization digital ID, desktop</td>
<td>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.</td>
</tr>
<tr>
<td>organization digital ID, server</td>
<td>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.</td>
</tr>
<tr>
<td>PKCS#11 device</td>
<td>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).</td>
</tr>
<tr>
<td>PKCS#11 digital ID</td>
<td>An ID on a PKCS# device. A device may contain one or more IDs.</td>
</tr>
<tr>
<td>PKCS#11 format</td>
<td>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.</td>
</tr>
<tr>
<td>PKCS#11 module</td>
<td>The software module that drives a PKCS#11 device.</td>
</tr>
</tbody>
</table>
## Table 81 Security Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKCS#11 token</td>
<td>See PKCS#11 device.</td>
</tr>
<tr>
<td>PKCS#12</td>
<td>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.</td>
</tr>
<tr>
<td>PKCS#7</td>
<td>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.</td>
</tr>
<tr>
<td>privileged context</td>
<td>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.</td>
</tr>
<tr>
<td>qualified certificates</td>
<td>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.</td>
</tr>
<tr>
<td>qualified electronic signatures</td>
<td>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.</td>
</tr>
<tr>
<td>roaming ID</td>
<td>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.</td>
</tr>
<tr>
<td>root certificate</td>
<td>The top-most certificate in a certificate chain. It is sometimes used as a trust anchor.</td>
</tr>
<tr>
<td>secure signature-creation devices</td>
<td>(SSCD) Software or hardware products used to store and apply signature code and that are designed for qualified electronic signatures</td>
</tr>
<tr>
<td>security restricted property or method</td>
<td>A property or method whose availability is restricted to certain events such as batch processing, console execution, or application startup. For example, in Acrobat 7.0, a security-restricted method (S) can only be executed through a menu event if one of the following is true: The JavaScript user preferences item “Enable menu items JavaScript execution privileges” is checked or the method is executed through a trusted function. The JavaScript for Acrobat API Reference identifies the items that have restrictions.</td>
</tr>
<tr>
<td>SSCD</td>
<td>See Secure signature-creation devices</td>
</tr>
<tr>
<td>timestamp</td>
<td>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).</td>
</tr>
<tr>
<td>trust anchor</td>
<td>A certificate in a certificate chain that is trusted for selected operations. It could be an ICA 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 CA.</td>
</tr>
</tbody>
</table>
Index

.msc 139
.msi 139
.msp 139
.mst 139
.ocx 139
.ops 139
.p12 168
.p7b 168
.p7c 168
.pcd 139
.pfx 168
.pif 139
.prf 139
.prg 139
.pst 139
.rar 139
.reg 139
.scf 139
.scr 140
.sct 140
.sea 140
.shb 140
.shs 140
.sit 140
.tar 140
.tgz 140
.tmp 140
.url 140
.vb 140
.vbe 140
.vbs 140
.vsmacros 140
.vss 140
.vst 140
.vsw 140
.webloc 140
.ws 140
.wsc 140
.wsf 140
.wsh 140
.zip 140
.zlo 140
.zoo 140

1
1000 69
1001 69
1002 69
1003 69
1004 69
1005 69
1006 69
1007 69
Acrobat 8 Family of Products
Security Feature User Guide

1008 69
1009 69

2
2004 70
2006 70
2007 70
2009 70
2010 70
2011 70
2012 70
2013 70
2014 70

3
3000 70
3001 70
3002 70
3003 70

4
4000 71
4001 71
4002 71

A
Access (Microsoft) 139
Access Add-in (Microsoft), MDA Access 2 Workgroup (Microsoft) 139
Access Add-in Data (Microsoft) 139
Access Data Access Page (Microsoft) 139
Access Diagram Shortcut (Microsoft) 139
Access Macro Shortcut (Microsoft) 139
Access MDE Database file (Microsoft) 139
Access Module Shortcut (Microsoft) 139
Access Project (Microsoft) 138
Access Project Extension (Microsoft) 138
Access Query Shortcut (Microsoft) 139
Access Report Shortcut (Microsoft) 139
Access Stored Procedures (Microsoft) 139
Access Table Shortcut (Microsoft) 139
Access Type 41
Access View Shortcut (Microsoft) 139
Access Wizard Template (Microsoft) 139
Access Workgroup Information (Microsoft) 139
Actions that can be associated with a signature field 67
Active Server Page 138
adbe.pkcs7.detached 121
adbe.pkcs7.sha1 121
adbe.x509.rsa_sha1 121
Add Digital ID dialog 14
Adding a Certificate From a Signature 33
Adding a Roaming ID Account to Get a Roaming ID 27
Adding an ID that Resides on External Hardware 28
Adding and Removing Digital ID Files from the File List 20
Adding Custom Signing Reasons 117
Adding Files to the Black and White Lists 140
Adding Someone to Your Trusted Identity List 33
Adobe Profile Files 168
Allowing and Blocking Specific Web Sites 144
Allowing Attachments to Launch Applications 141
Alternate certificate URL seed value 128
An early compressed file format 140
Approval Signature 73
approval signature 168
Arranging Signature Fields 62
Associating a Certificate with a Contact 45
Attachment
Cannot open warning 138
Dangerous type warning 137
Attachment panel in Trust Manager 141
Authoring a Document with Multiple Fields 63
Authoring Signable Documents 48
Authoring Signable Forms 63
Automating signing tasks 129

B
BASIC Source Code 138
Batch Processing 138
Before You Sign . . . 73
Black Lists and White Lists 137
Blocked URL alert 143
Bzip 2 UNIX Compressed file (replaces BZ) 138
Bzip UNIX Compressed file 138

C
CA 168
CDS 168
CDS digital ID 168
CDS digital ID certificate 168
certificate authority (CA) 168
Certificate Chains and Trust Anchors /Roots 158
Certificate file 138
Certificate issuer and subject seed value 125
Certificate key usage seed value 126
Certificate policy seed value 127
Certificate Revocation List (CRL) 168
Certificate Revocation List. 168
Certificate Trust Settings 36
Certificate trust settings 37, 38, 156
Certificate Viewer 103
Certificate viewer
Trust tab 94
Certificate Viewer information 102
Certificates
Contact Information 148
Selecting a digital ID for export 164
Verifying originator 104
certificates 168
Certificates in the Trusted Identities list 161
Certification Signature 73
certification signature 168
Certification Workflow for Documents with Multiple Signers 76
certified document 168
Certified document indicators 76
Certified Document Services (CDS) 168
Certified Document Services. 168
certify or certifying 168
Certifying a Document 78
Certifying a document
  Document integrity warnings 79
  Signature 80
certspec 113
Changes Across Releases 113
Changing a PKCS#12 File's Password Timeout 21
Changing a Trusted Identity's Certificate Association 45
Changing an ID File's Password 20
Changing Passwords 29
Changing the Default Field Appearance 61
Changing the Default Signing Method 57
Check revocation 102
Checking Certificate Revocation Status 104
Checking Document Integrity 68
Clearing One or More Signatures 83
Command 138
Comment or form field may silently change 69
Compare
  By page 110
  By page summary report 109
  Choose compare report type 109
  Displaying changed text 111
  Select file drop-down menu 108
  Type of comparison 108
Comparing a Signed Version with the Current Version 107
Comparing Documents 107
Comparing Documents by Page 109
Comparing Documents by Word 110
Compiled HTML Help 138
Compressed archive (LH ARC) 139
Compressed Archive file 140
Compressed archive of Mac files (Stuffit) 140
Configure Document Behavior 77
Configuring Multimedia Trust Preferences 133
Contacts
  Deleting 47
  Selecting certificates 45
  Viewing details 44
Controlling Access to Content Outside a PDF 142
Controlling Document Behavior with Seed Values 112
Controlling Multimedia 132
Controlling Multimedia in Certified Documents 134
Creating a Blank Signature Field 58
Creating a Custom Signature 53
Creating a Custom Signature Appearance 53
Creating a Custom Watermark or Background 53
Creating a Self-Signed Digital ID 22
Creating Multiple Copies of a Signature Field 62
Creation time 89
CRL 168
Current time 89
Custom signature appearance 52
Custom Signature Appearances 52
Custom Workflows and Beyond 129
Customizing a Digital ID Name 17
Customizing Field Appearances 60
Cut, Copy, and Paste Signature Fields 61
D
Default Behavior
  Black and White Lists 137
Default prohibited file types 138
Deleting a Certificate 47
Deleting a Directory Server 42
Deleting a PKCS#12 Digital ID 25
Deleting a Windows Digital ID 26
Deleting Contacts and Certificates 46
Details 102
digestMethod 113
Digital ID
  Certificate viewer 19
  Components 12
  Configuration 24
  Deleting 25
  From others 12
  ID export options 162
  Managing trusted identities 32
  PKCS#12 location and password 25
digital ID 169
Digital ID Basics 11
Digital ID Directory servers
  Email details 166
  Export destination 165
  Importing 152
  Sender's identify 165
  Server list 41
  Setting defaults 43
  Setting server details 42
Digital ID files
  Password configuration 21, 30
  Timeout settings 22
Digital ID Files menu 19
Digital ID format selection 23
Digital ID Management 14
Digital ID Related Files and Storage Mechanisms 12
Digital ID-related file types 13
Digital IDs
  Searching the Windows Certificate Store 35
digital signature 169
Digital signature icons 86
Digital Signature Properties
  Document Versioning panel 95, 99, 106
  Modifications panel 107
Digital Signature Validation 84
Digital Signatures Advanced Preferences
  Signature verification 88
Digital signatures user interface 74
digitally sign 169
Directory Name 41
Displaying Someone’s Certificate in the Certificate Viewer 102
Distributing a Trust Anchor or Trust Root 158
Document contains hidden behavior 69
Document contains links to external PDFs 69
Document Legal Defensibility 75
Document links to external content 70
Document Locking 75
Document may not open in the future 70
Document may silently launch menu items  69
Document Message Bar
  Level A compliance  71
  Level B compliance  71
Documentation related to Acrobat security  10
DOS CP/M Command file, Command file for Windows NT  138
Downloaded roaming ID certificates  155

E
Edit Contact dialog  44, 46
Editing Directory Servers Details  42
Editing or Deleting a Signature Appearance  55
  EE  169
electronic signatures  169
Emailing a certificate request  163
Emailing a digital ID  162
Emailing Certificate or Contact Data  44
Emailing Server Details  164
Emailing Your Certificate  161
Emailing your certificate  147
embedded JavaScript  169
embedded validation response  169
Embedding Revocation Information in a Signature  122
Enabling JavaScript to Set Seed Values  122
end entity certificate (EE)  169
end entity certificate.  169
Executable Application  138
Executable file  138
Execute a Menu Item  67
Export the Trust Anchor  159
Exporting a Certificate Other than Yours to a File  105
Exporting Application Settings and Digital ID Data  157
Exporting Server Details  166
Exporting Your Certificate  161
External connection warning  144
External Content  70
External Content and Document Security  132

F
FDF Email attachment  146
filter  113, 120
Finding a Digital ID in a Windows Certificate Store File  26
Finding an Existing Digital ID in a PKCS#12 File  19
flags  113, 114, 123
Forcing a Certification Signature and Document Locking  115
FoxPro Compiled Source (Microsoft)  138

G
Generic ID Operations  15
GeoTrust  169
Getting and Using Your Digital ID  11
Go to 3D View  67
Go to a Page View  67
Green Check and Caution Triangle  99
Gzip Compressed Archive  138

H
Hash algorithm seed value  122
Hidden  60
Hidden but printable  60
High Privilege JavaScript Defined  135
How Do I Validate a Timestamp in a Signature?  96
Hypertext Application  138

I
Identity preferences  15
IIS Internet Communications Settings (Microsoft)  138
IIS Internet Service Provider Settings (Microsoft)  138
Import Form Data  67
Importing a Certificate From a File  34
Importing a Trust Anchor and Setting Trust  155
Importing Adobe Policy Server Settings  152
Importing and Exporting Directory Server Settings  43
Importing Application Settings and Digital ID Data  146
Importing APS settings  153
Importing digital ID data  34
Importing Directory Server Settings  151
Importing Multiple Certificates  149
Importing multiple certificates  149
Importing Roaming ID Account Settings  154
Importing roaming ID server settings  154
Importing Someone’s Certificate  148
Importing Timestamp Server Settings  150
individual digital ID  169
Information or Setup file  138
Initialization/Configuration file  138
Interactive feature warnings  69
intermediate certificate authority (ICA)  169
Internet Document Set, International Translation  138
Internet Location  140
Internet Security Certificate file (MIME x-x509-ca-cert)  138
Internet URL Access  142
issuer  123

J
Java Class file  138
Javascript and Certified Documents  136
JavaScript Security option  136
JavaScript Source Code  139
JScript Encoded Script file  139

K
keyUsage  124

L
Launch Attachment dialog  141
Legal Notice  102
legalAttestations  113, 114
Local time vs Timestamp Time  95
Locking Fields Automatically After Signing  64
Logging in to a Device  30
Logging in to a Roaming ID Account  27
Logging in to a roaming ID server  155
Logging in to an Adobe Policy Server 153
Logging in to PKCS#12 Files 22

M
Mac OS Command Line executable 138
Mac OS Finder Internet Location 140
Macintosh BinHex 2.0.2 file 138
Macintosh BinHex 4 Compressed Archive 138
Making a contact a trusted identity 150
Making a Field a Required Part of a Workflow 65
Manage Internet Access dialog 143
Manage Trust for Multimedia Content dialog 132
Managing Contacts 43
Managing IDs Accessible via PKCS#11 Devices 28
Managing PKCS#12 Digital ID Files 19
Managing Roaming ID Accounts and IDs 26
Managing Windows Digital IDs 26
Manually Configuring a Directory Server 41
Maximum Number of Records to Receive 41
mdp 113, 114
Media Attachment Unit 139
Microsoft Management Console Snap-in Control file
(Microsoft) 139
Microsoft Object Linking and Embedding (OLE) Control
Extension 139
Mouse Down 66
Mouse Enter 66
Mouse Exit 66
Mouse Up 66
MS Exchange Address Book file, Outlook Personal Folder file
(Microsoft) 139
Multimedia behavior workflow 133
Multimedia Trust Manager 134

N
Name 60
No Compliance with PDF/SigQ 72

O
OCSP 169
Office Profile Settings file 139
oid 124
On Blur 66
On Focus 66
Online Certificate Status Protocol (OCSP) 169
Open a File 67
Open a Web Link 67
Optional
  Setting Certificate Trust Level 160
organization digital ID, desktop 169
organization digital ID, server 169
Orientation 60

P
Page content may silently change 70, 71
Password 41
PDF content contains errors 71
PDF Content with variable rendering 70
PDF/SigQ 68
PDF/SigQ Checker Results 69
PDF/SigQ Conformance Report dialog
  Level B compliance 72
  No compliance with level A or B 72
PDF/SigQ Level A Compliant 71
PDF/SigQ Level B Compliant 71
Personalizing an ID name 18
PKCS#11 device 169
PKCS#11 digital ID 169
PKCS#11 format 169
PKCS#11 module 169
PKCS#11 Security Settings menu items 29, 30
PKCS#11 token 170
PKCS#12 170
PKCS#7 170
Play a Sound 67
Play Media (Acrobat 5 Compatible) 67
Play Media (Acrobat 6 Compatible) 67
 Policies 102
Policy OID 127
Policy restrictions 39, 40, 157
Port 41
Presentation elements may change appearance 69
Preventing Multimedia Playback in Certified Documents 135
Preview document mode preference 50
Preview Mode and PDF/SigQ Conformance 68
privileged context 170
Problems encountered 102
Program file 139
Provide Instructions to the Trusted Root Recipients 160

Q
qualified certificates 170
qualified electronic signatures 170
Question Mark and Caution Triangle 99
Question Mark and Pen 100
Question Mark and Person 100

R
Read an Article 67
Read Only 60
Reason field behavior 118
reasons 113, 114
Red X and Pen 101
Registering a Digital ID for Use in Acrobat 13
Registration Information/Key for Windows 95/98, Registry
  Data file 139
Requesting a Certificate via Email 163
Requesting a Digital ID via Email 34
Required 60
Required field not signed alert 65
Reset a Form 67
Resetting the Black and White Lists 141
Resource access 142
Responding to an Emailed Request for a Digital ID 146
Restricting Signing to a Roaming ID 128
Revalidate signatures warning 97
Revocation 102
Roadmap to Security Documentation 9
roaming ID 170
Roaming ID Security Settings menu items 28
Roaming ID seed value 129
Roaming ID server name and URL 154
root certificate 170
Run a JavaScript 67
Running JavaScript and Dynamic Content 87

S
Saving Certificate or Contact Details to a File 44
Saving Your Digital ID Certificate to a File 162
Search Base 41
Searching for a document recipients 36
Searching for Digital ID Certificates 35
secure signature-creation devices 170
Secure time 89
security restricted property or method 170
Security Settings Console 16
Security settings menu and manager 14
Security Settings menu items 165
Security Terms 168
Seed value
  Custom signing reason 118
  Forcing a certification workflow 116
  mdp 117
  Reason not allowed error 119
  Specifying certificates for signing 123
  Specifying signature components 121
Seed Value Basics 112
Seed value changes across releases 113
Seed values
  certSpec properties 123
  Custom legal attestations 117
  JavaScript debugger 115
  timeStampspec properties 119
SeedValue object properties 113
Selecting a certificate chain for export 159
Selecting a digital ID 147
Self-expanding archive (used by Stuffit for Mac files and possibly by others) 140
Server Name 41
Set Layer Visibility 67
Setting Certificate Trust 37
Setting Digital Signature Validation Preferences 88
Setting High Privilege JavaScript Options 135
Setting Signing Preferences 49
Setting up a Document for Certification 77
Setting up a Timestamp Server 56
Setting Up Automatic Signature Validation 87
Setting up Form Fields 77
Setting up the Signing Environment 48
Setting up Your Environment for Signature Validation 87
Sharing (Exporting) a Digital ID Certificate 16
Sharing Settings & Certificates with FDF 145
Shell Scrap Object file 140
shouldAddRevInfo 113, 114
Show/Hide a Field 67
Signature appearance
  Configuration 55
  New button 54
Signature components 85
Signature creation preferences 50
Signature field
  Action properties 66
  Appearance properties 61
  Default appearance 59
  Edit options 62
  General properties 59, 60
  Multiple copy options 63
  Signing properties 64
Signature field sign menu 81
Signature Properties
  Summary 93
Signature Status 85
Signature Types 73
Signature validation confirmation 92
Signature Validity Basics 84
Signatures tab
  Validate signature 91
Signer Details 102
Signing a document
  Signature details 82
Signing an Existing Field with a Digital Signature 73
Signing an FDF file 158
Signing Basics 73
Signing Documents in Acrobat 81
Signing environment preferences 49
Signing in a Browser 83
Signing User Interface 74
Signing With a Certification Signature 74
Signing with a Timestamp 55
Signing with an Approval Signature 81
Specifying a Default Directory Server 43
Specifying a Post-Signing Action 65
Specifying a Signature Hash Algorithm 121
Specifying a URL When a Valid Certificate is not Found 128
Specifying Alternate Signature Handlers and Formats 120
Specifying Certificate Properties for Signing 122
Specifying Certificates by Key Usage 126
Specifying Certificates by Policy 127
Specifying Digital ID Usage 16
Specifying General Field Properties 59
Specifying Signing Certificates Origin 125
Specifying Timestamps for Signing 119
SSCD 170
subFilter 113, 114
subfilter 120
subject 124
subjectDN 124
Submit a Form 67
Summary 102
Supported Seed Values 113

T
Tape Archive file 140
Temporary file or Folder 140