



PDF Creation Date:

May 5, 2008

Digital Signature Appearances

Adobe Acrobat

Version 9.0

© 2006-2008 Adobe Systems Incorporated. All rights reserved.

Adobe® Acrobat® SDK Digital Signature Appearances for Microsoft® Windows®, and Mac OS®.

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

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

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

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

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

Mac OS is a trademark of Apple Computer, Inc., registered in the United States and other countries.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are the property of their respective owners.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

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

Contents

1	Digital Signature Appearances.....	4
	Signature interoperability	4
	Annotation appearances	4
	Example signatures	5
	Standard AP dictionary and layers	5
	Assembling XObject layers	6
	Top-level XObject	6
	Second-level XObject	6
	Layer n0	7
	Layer n2	7
	Signature annotation with no appearance	7
	Signature objects	8
	DigSigHFT appearance APIs	9
	Appearance management with PubSec	10
	Appearance file	10
	Annotation appearance.....	11
	Appearance watermarks	11
	Creating a watermark.....	12
	Testing or correcting a watermark file	13

Digital Signature Appearances

This document provides guidelines for the programmatic generation of standardized appearances for signatures in PDF documents. It contains information for developers as well as system administrators who provide support for signature workflows.

Signature interoperability

The digital signature API in Acrobat allows plug-ins to implement *digital signature handlers* to create and verify document signatures.

The goal is to achieve file interoperability between handlers, particularly for those that are implemented based on public-key cryptography. This interoperability allows a PDF file that is signed with one digital signature handler to be verified with a different digital signature handler, assuming the other handler understands the same basic signature format.

Interoperability required a standard PDF syntax for implementation of signatures. This standard syntax has two primary components:

The *signature dictionary* which stores the actual signature. It includes attributes such as the signer's name, the signing time, the signed hash of the file, and the signer's certificate (for example, embedded as a PKCS#7 object). Standard syntax for the signature dictionary is defined using the `SubFilter` attribute of the signature dictionary. For more information, see the *PDF Reference*.

The *signature appearance* which is the user-visible representation of the signature. The signature appearance is specified in the signature annotation. A signature appearance is optional and standardizing the appearance is not necessary for the cryptographic purpose of verifying a signature. However, the user of an appearance can facilitate a consistent and intuitive user experience.

Annotation appearances

Acrobat digital signatures exist as signature form fields in a PDF file. As with any form field, signature form fields are associated with an annotation dictionary. Appearances for any annotation, including signatures, are contained in the appearance dictionary, or AP attribute, of the annotation. Follow the guidelines in this document when creating a signature appearance creation to ensure the best operation within Acrobat.

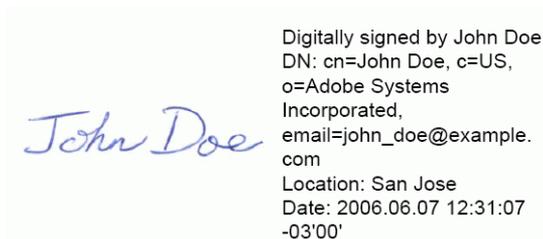
When developing plug-ins, make use of the complete signature appearance support that is part of the PubSec layer of code that resides between the Acrobat PubSecHFT and the DigSigHFT. For more information, see ["Appearance management with PubSec" on page 10](#).

Example signatures

These example signatures are referred to later in this document.

Tip: Signature status icons may or may not appear depending on your version of Acrobat and how it is configured. For example, in Acrobat 9.0, icons appear in the signature pane rather than in the signature field. Also note that icon styles have changed across past versions of Acrobat and Adobe Reader.

Example: *Signature with no watermark*

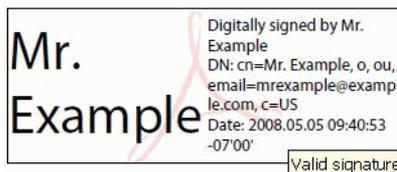


Example: *PubSec generated signature appearance with watermark Acrobat*



Example: *PubSec generated signature appearance Acrobat with overlays*

Tooltip overlays are a common usability feature that make it easier for signature validators to determine signature status. Product versions prior to 9.0 also display a signature status icon. After 9.0, the status icon only appears in the signature pane.



Standard AP dictionary and layers

A signature's appearance is contained in the N (Normal) attribute of the signature annotation's AP dictionary. The R (Rollover) and D (Down) attributes are not used. The appearance is contained in a number of XObjects that are assembled to create layers. These XObjects are pulled together with the Do page marking operator as described in ["Assembling XObject layers" on page 6](#).

Signature appearances use two standard layers and can have a variable number of optional user-definable layers. Each layer is represented by its own XObject.

Note: Prior to Acrobat 6.0, signature appearances were manipulated at run-time in order to display the validity of the signature. The validity was shown as a graphic icon and with an additional, optional text message. The manipulated portions of the signature appearance were contained in layers n1, n3 and n4. Beginning with version 6, Acrobat does not maintain support for signature appearances that can be manipulated, though legacy signatures with these appearances may continue to display correctly. Use of layers n1, n3, and n4 is not recommended.

The standard XObject layers are as follows:

n0: The background layer that is preset when creating the signature field (such as when using the Acrobat form creation tool). In the example appearances in [“Example signatures” on page 5](#), this layer is blank.

n2: The signature appearance layer that contains information about the signature. In the example [Signature with no watermark](#), this is the layer that contains the line art for the handwritten signature and the text giving the signer’s name, date, reason, and location. The content of this layer can be dynamically created when the signature is created, but thereafter it remains static. Specifically, it remains static when the validity state is changed.

The layers are painted in sequence, beginning with n0.

Handlers can define their own custom layers. Custom layers must not use the names n0 through n4.

Assembling XObject layers

An appearance is assembled with the top-level and second-level XObjects and the standard layers. Care should be taken to follow the recommendations for setting the matrix values to indicate unity transforms.

Top-level XObject

The AP dictionary’s N attribute references a top-level XObject. This top-level XObject is necessary to properly resize signature appearances when these appearances are referred to by more than one signature field. For an example, see object 17 in the graphic [“Signature appearance objects in a PDF file” on page 9](#).

The top-level XObject performs a Do on the second-level XObject as follows:

```
q 1 0 0 1 0 0 cm /FRM Do Q
```

The in-line cm matrix may change if the signature is resized. The Matrix is unity (x,y translation of zero, scale factor = 1), and the BBox is the current bounding box of the annotation.

The location of a signature within a document can have a bearing on its legal meaning. For this reason, signature fields never refer to more than one annotation. If more than one location is associated with a signature, the meaning may become ambiguous. Regardless, all signatures should include the top-level XObject.

Second-level XObject

The standard layer n0 appears by name in the Resource dictionary of the second-level XObject’s dictionary. The Matrix of this XObject is unity and the BBox is that of the original annotation.

The XObject stream appears as follows:

```
q 1 0 0 1 0 0 cm /n0 Do Q
```

Layer n0

This layer renders the background and border of the annotation. The attributes that specify the background and border exist in the signature field dictionary. These attributes are not set if the signature is created on-the-fly by the plug-in. However, if Acrobat was used to create a blank signature field, these values are set.

The Acrobat forms HFT calls can help in the process of obtaining these values. For example, to extract border and background information, refer to `AFPDWidgetGetAreaColors` and `AFPDWidgetGetBorder`. Other Acrobat forms HFT calls create the n0 XObject. These API calls are documented in the *Acrobat and PDF Library API Reference*.

Note: For information on the DigSigHFT calls that can be used to create appearances, see [“DigSigHFT appearance APIs” on page 9](#).

If you are using a blank n0 layer, use this call to obtain the blank object:

```
DigSigGetStdXObject ( cosDoc, DSBlankXObject ) ;
```

Layer n2

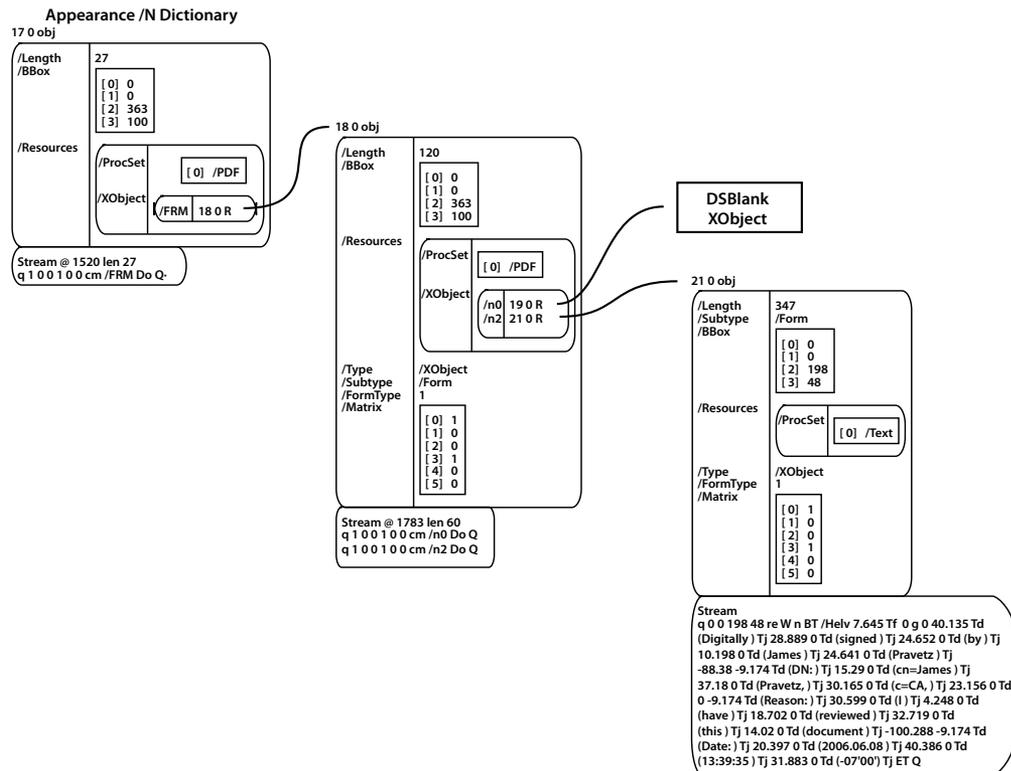
The handler can create any appearance for this layer. The layer is static. It is not changed when the validity state of the signature is changed.

All appearance handlers that render text honor the font type and color defaults that were set for the signature annotation. These defaults can be obtained by calling `AFPDFieldGetDefaultTextAppearance` as defined in the *Acrobat and PDF Library API Reference*.

Signature annotation with no appearance

PDF signatures can be visible or invisible. In both situations there is an associated annotation. For invisible signatures, the annotation has a rectangle array with values [0 0 0 0]. This annotation is usually attached to the viewed page when the signature is created. Despite not having an appearance, the annotation AP and N dictionaries may be present in some versions of Acrobat. If present, N references the `DSBlankXObject` (blank) XObject.

Signature appearance objects in a PDF file



DigSigHFT appearance APIs

Several API calls in the DigSigHFT make the generation of signature appearances easier for third-party developers. These procedures also make it easier to create signatures that conform to the layered XObject approach. The procedures are declared in DigSigHFT.h and documented in the [Acrobat and PDF Library API Reference](#). The calls are:

DigSigCreateStdXObj — Creates a standard XObject from one of the values in the enum DSXObjType. This call allows a handler to get the predefined XObject for either a signature:

A signature in a document which has not changed but for which the signer's certificate cannot be validated (the signature icon is a question mark that covers a pen); or,

A signature on a document that has not changed and which has a validated certificate (the signature icon is a green check mark).

DigSigAPCreateLayeredStream — Creates the N dictionary and lower level XObjects given the XObjects for the various layers. The layers have defaults, so you do not need to define them.

DigSigAPXObjectFromXObjList — Creates a single XObject from multiple components. Use this routine to combine graphics and text components to make layer n2. Controls are given for placement of the graphics and text within the annotation.

DigSigAPXObjectFromLogo — Creates an XObject from a string that contains the marking operators needed to draw an object. Adobe's signature handlers use this routine to create the Acrobat

logo that is used as a signature watermark. As an example, the string that is used to create the logo is created by concatenating the strings shown in the following example.

DigSigAPCreateCompositeTextXObject — Combines various text elements into a single text XObject. This call calls `DigSigAPXObjectFromXObjectList` to do the actual stitching of text elements.

Example: Example string to use with DigSigAPXObjectFromLogo

```
// Precise bbox of Acrobat logo stream stmLogoLogo
const ASFixedRect CSSEngine::stmLogoBBox =
{ 0,0,0x00630000,0x00630000 }; // 0,0,99,99
// Stream data for Acrobat logo
const char* const CSSEngine::stmLogoData[] = {
"% Acrobat Logo",
"q",
"0 J 0 j 4 M [] 0 d",
"1 0 0 1 0 -1 cm",
"1 0.85 0.85 rg", /* faded red */
"0 i",
"96.1 24.7 m",
"96.4 24.7 l",
"96.8 24.7 97 24.6 97 24.3 c",
.
.
.
.
.
"51.4 45.2 60.3 39.4 65.1 36.6 c",
"53.5 34.6 40.1 31.1 28.1 25.9 c",
"h f",
"Q",
NULL
};
```

Appearance management with PubSec

The authoring of public key security handlers involves using common signature appearance code in the PubSec software module. PubSec is a layer of code between the DigSigHFT and the PubSecHFT. Plug-in authors can allow PubSec to take care of all signature appearance generation and maintenance, but they may also continue to create their own signature appearances.

For more information on the PubSec API, see the *Acrobat and PDF Library API Reference*.

Appearance file

PubSec maintains a user-configurable database of signature appearances that can be used when signing a document. Users configure signature appearances using the preferences settings in Acrobat. The PubSec signing dialog boxes access this same database, thereby enabling end users to configure and choose the which signature appearance to use. The database is a PDF file called `appearances.acrodata` that, for Windows, is stored in the Application Data section of a user's file system. An example path is:

```
C:\Documents and Settings\\Application Data\Adobe\Acrobat\\Security\
appearances.acrodata
```

The PubSec signature appearance database is accessible via the `DSAPFile` interface of PubSecHFT. The `DSAPFile` interface exposes enough calls to allow maintenance of the database. Maintenance includes enumeration, addition, modification, and deletion calls:

DSAPFileAcquire — Acquires the `DSAPFile` object for PubSec's signature appearance database file. If the file has not already been acquired, it will be opened. PubSec acquires the database file whenever it is needed by PubSec, so handlers do not need to acquire this file unless it needs to access the file for other reasons.

DSAPFileRelease — Releases the file, with the potential to close and save the file.

DSAPFileSave — Saves the file if it is dirty, leaving the file open.

DSAPFileGetCount — Returns the number of configured appearance entries in the appearance database file.

DSAPFileCanDeleteNthEntry — Returns true if this entry in the `DSAPFile` can be edited (that is, is not read-only). Index values that are less than zero correspond to the default appearance, which is called the `StandardText` appearance in the user interface.

DSAPFileGetNewNthName — Gets a copy of the name of the Nth appearance object in the file. Use this when building a list of appearances for a user to choose from or edit.

DSAPFileRemoveNthEntry — Deletes the Nth appearance entry from the file.

DSAPFileEditNthEntry — Displays a user interface that allows the user to edit the Nth entry of the appearance file. Passing in indexes larger than the number of entries in the appearance file will create a new entry in addition to displaying the aforementioned user interface.

DSAPFileCopyNthEntry — Creates a copy of the Nth entry in the appearance file and appends the copy to the end of the list of signature appearances in the file. When copying default appearance entries, the resulting copy will not be considered a default appearance entry.

Annotation appearance

Signatures that are created by PubSec use the layered XObject approach.

Example: *Signature appearance*

```
q 1 0 0 1 0 0 cm /n0 Do Q  
q 1 0 0 1 0 0 cm /n2 Do Q
```

PubSec correctly displays legacy signatures that use the n1, n3, and n4 layers, but support of these layers in future versions of Acrobat may be discontinued at any time. At run time, PubSec changes these layers to show the validity state. PubSec looks for the presence of these layers to determine if run-time appearance modification is required. In some situations, PubSec will specifically warn against the presence of signatures with these layers because, for conformance with various signed document standards, signature appearances should be static.

Appearance watermarks

PubSec allows customization of the watermark that is shown as part of a signature appearance. Customization can be done by a PubSec handler or by a system administrator using the `SignatureLogo.pdf` file, which is described below.

The watermark is incorporated as part of the n2 layer of the signature appearance. In the example [PubSec generated signature appearance with watermarkAcrobat](#), layer n2 is composed of the watermark, which is

the logo from the *US Department of the Interior*, the text "Digitally signed by ...", and the signer's name, Inky Pinky.

Creating a watermark

To customize the watermark, a system administrator can place a file called `SignatureLogo.pdf` in the same folder where the `appearances.acrodata` file is stored. In Windows, an example path to the appearance file is:

`C:\Documents and Settings\user\Application Data\Adobe\Acrobat\x.0\Security\appearances.acrodata`

If this file exists, the first page of this file is extracted and used for the watermark. If this file does not exist, PubSec handlers can provide the Acrobat logo as a watermark using the PubSec API `PSGetLogo` procedure call. If the `PSGetLogo` procedure is `NULL`, no logo watermark is added to the signature appearance.

Creating a watermark from a vector image

1. Open a file containing the design (for example, a company logo) using Adobe Illustrator®.
2. Isolate the design and copy it to the clipboard.
3. Create a new document and paste the design into this new document.
4. Select all. Note that if your logo has objects that are drawn over top of other objects, the underlying objects may become visible through the higher layers. In this situation you may have to group the overlapping objects before proceeding to the next step.
5. Click **Window > Transparency** and drag the transparency slider to approximately 20%. Note that you want the appearance to be sufficiently subdued, so a value of less than 20% may be necessary for some designs.
6. Click **Object > Flatten Transparency**, set the Raster/Vector balance to 100%, and click OK.
7. Save the file as a PDF document.
8. Open the PDF file in Acrobat.
9. Click **Document > Crop Pages**. Check the remove white margins option and click OK.
10. Save the file in an appropriate location.
11. Test the file (see ["Testing or correcting a watermark file" on page 13](#)).

Creating a watermark from a vector image

1. Drag the bitmap image file to Acrobat to create a new PDF file.
2. Save the PDF file.
3. Prepare the file using the commenting stamp tool (see ["Testing or correcting a watermark file" on page 13](#)).

Testing or correcting a watermark file

After you have created a watermark, you can test it or correct it if it causes problems.

Testing a watermark PDF file:

4. Copy or save your watermark PDF file as SignatureLogo.pdf in the Security folder.
5. In Acrobat, click **Edit > Preferences > Security**.
6. Choose **New** to create a new signature appearance.
Your watermark should appear in the Preview box.
7. Set the desired options.
8. Choose **OK** until the dialog closes.
9. Try signing a document to verify that the watermark is working. If you receive an error when testing a document (or if the watermark fails to appear in the Preview box), see .

Note: Not all PDF files can be used as signature watermarks. If the structure of the first page is not correct, you can correct it with the Stamp tool in Acrobat.

Correcting the structure of a watermark file:

10. Note the contents of the Stamps folder to ensure you do not overwrite any existing stamps. This folder is at the same level as the Security folder, for example:
C:\Documents and Settings\user\Application Data\Adobe\Acrobat\x.0\Stamps
C:\Documents and Settings\<user>\Application Data\Adobe\Acrobat\<version>\Stamps\
11. In Acrobat, click **Tools > Comment & Markup > Stamps > Create Custom Stamp**. Select your watermark file and import it as a new stamp in a new category.
12. Copy the new file that has been added to the Stamps folder to the Security folder and rename it as SignatureLogo.pdf.
13. Test the file (see the preceding procedure).

Adobe does not provide support for logo watermark preparation or use. If you have trouble with any aspects of this procedure, consult a graphics professional.