Developer FAQ

Adobe® Acrobat® SDK

November 2006

Version 8.0

Adobe® Acrobat® SDK 8.0 Developer FAQ for Microsoft® Windows®, Mac OS®, Linux® and UNIX®

Edition 1.0, November 2006

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

Contents

2 Understanding the Acrobat SDK (Continued)

Preface

This document addresses frequently asked questions about the Adobe® Acrobat® Software Development Kit (SDK).

What's in this guide?

This guide provides answers to frequently asked questions about the Acrobat SDK. It includes discussion about the Acrobat family of products and supporting technologies, introduces the various aspects of the Acrobat SDK and its uses, discusses requirements surrounding forms, provides information on licensing and technical limitations, provides information on user and developer resources, highlights information relevant to development for Adobe Reader®, and provides information on enterprise installation.

Who should read this guide?

This guide is meant for developers who would like to understand how the Acrobat SDK can be used with Adobe Acrobat or Reader. For most uses of the Acrobat SDK, a programming or scripting background is assumed.

Related documentation

To help you get started with the Acrobat SDK, see the following documents.

For information about	See
A guide to the documentation in the Acrobat SDK.	*Acrobat SDK Documentation Roadmap*
Known issues and implementation details.	*Readme*
Answers to frequently asked questions about the Acrobat SDK.	*Developer FAQ*
New features in this Acrobat SDK release.	*What's New*
A general overview of the Acrobat SDK.	*Overview*
A guide to the sections of the Acrobat SDK that pertain to Adobe Reader.	*Developing for Adobe Reader*
A guide to the sample code included with the Acrobat SDK.	*Guide to SDK Samples*
Prototyping code without the overhead of writing and verifying a complete plug-in or application.	*Snippet Runner Cookbook*
Configuring and administering a system for online collaboration using comment repositories, Acrobat and Adobe Reader.	*Acrobat Online Collaboration: Setup and Administration*

For information about	See
Enabling Acrobat to save documents in a customized text-based format.	*Extending the Acrobat SaveAsXML Plug-in*
Using DDE, OLE, Apple events, and AppleScript to control Acrobat and Adobe Reader and to render PDF documents.	*Developing Applications Using Interapplication Communication*
Detailed descriptions of DDE, OLE, Apple event, and AppleScript APIs for controlling Acrobat and Adobe Reader or for rendering PDF documents.	*Interapplication Communication API Reference*
Detailed descriptions of JavaScript™ APIs for adding interactivity to 3D annotations within PDF documents.	*JavaScript for Acrobat 3D Annotations API Reference*
Using JavaScript to develop and enhance standard workflows in Acrobat and Adobe Reader.	*Developing Acrobat Applications Using JavaScript*
Detailed descriptions of JavaScript APIs for developing and enhancing workflows in Acrobat and Adobe Reader.	*JavaScript for Acrobat API Reference*
Using RSS to track remote resources in an occasionally-connected environment.	*Acrobat Tracker*
Detailed descriptions of APIs for controlling Acrobat Distiller® for PDF file creation.	*Acrobat Distiller API Reference*
Specifying settings for the creation of PDF files.	*Adobe PDF Creation Settings*
A detailed description of an extension to the PostScript® language which allows the description of PDF features not found in standard PostScript.	*pdfmark Reference*
A detailed description of the PDF file format.	*PDF Reference*
Developing plug-ins for Acrobat and Adobe Reader, as well as for PDF Library applications.	*Developing Plug-ins and Applications*
Detailed descriptions of the APIs for Acrobat and plug-ins, as well as for PDF Library applications.	*Acrobat and PDF Library API Reference*
Detailed descriptions of the APIs for using assistive technology with PDF documents.	*PDF Accessibility API Reference*
Using JavaScript to perform repetitive operations on a collection of files.	*Batch Sequences*
A detailed description of the parameters for opening PDF files and for performing actions on them using a URL or command.	*Parameters for Opening PDF Files*
A list of the U3D elements supported by Acrobat.	*U3D Supported Elements*

1 Introduction

In this chapter, the Acrobat family of products and the Acrobat SDK are introduced, and developer support and licensing are discussed.

Basic Acrobat SDK questions

This section provides answers to basic questions related to the Acrobat family of products and the Acrobat SDK.

What Is Acrobat?

Adobe Acrobat 8.0 consists of a family of products for creating, modifying, indexing, searching, displaying, and manipulating Adobe PDF (Portable Document Format) files. For further information about the Acrobat family of products, see http://www.adobe.com/products/acrobat/main.html.

What is the Acrobat family?

- **Adobe Reader** for viewing, navigating, and printing PDF documents. Adobe Reader is free software that lets anyone view and print PDF files on all major computer platforms, as well as fill in and submit PDF forms. Adobe has distributed more than 525 million copies of the software worldwide. Reader 8 enables you to securely view, print, search, sign and verify the authenticity of PDF files. With Reader 8, you can easily collaborate on a document with the new "Start Meeting" conferencing button. Additionally, Reader 8 features a new simpler, streamlined interface with customizable tool bars.
- **Acrobat® Elements** for organizations requiring widespread PDF creation capability without the need to install the full Acrobat Distiller® product on every desktop. Acrobat Elements allows organizations to extend their investments in Microsoft Office by standardizing on PDF for document distribution. Acrobat Elements is available through licensing with minimum orders of 100 seats. There is no public API for Acrobat Elements.
- **Acrobat** for adding navigational links, annotations, and security options, in addition to the functionality provided by Adobe Reader. Acrobat 8.0 is offered in three configurations:
  - **Acrobat 8.0 Standard** — Offers all of the PDF creation and distribution tools in Acrobat Elements and enables business professionals to reliably create, combine, and control PDF documents for easy, more secure distribution and collaboration. Combine files from multiple applications into a single, polished PDF document. Protect sensitive information with passwords and permissions, and conduct collaborative document reviews by e-mail or server.
  - **Acrobat 8.0 Professional** — Contains all of the functionality of Acrobat 8.0 Standard and Acrobat Elements. It enables business professionals to reliably create, combine, and control PDF documents for easy, more secure distribution, collaboration, and data collection. Protect sensitive information with passwords, permissions, and digital signatures. Enable Adobe Reader users to participate in document reviews, fill and save forms, and digitally sign documents.
  - **Acrobat 8.0 3D** - Contains all of the functionality of Acrobat 8.0 Professional. It offers design engineers and technical publishers a complete and more secure way to collaborate with extended teams on 3D designs. Capture 3D designs from major CAD applications in PDF documents to provide rich, interactive 3D designs to users of free Adobe Reader software. Accelerate review cycles, streamline production, help reduce costly redesigns, and leverage existing CAD assets for technical manuals and marketing materials.

What are the tools for creating PDF files?

- The Adobe Acrobat family. Adobe Acrobat 8.0 Standard, Professional, and 3D include Acrobat Distiller for generating PDF files from PostScript files created with your favorite applications (including desktop publishing software).
- Acrobat Capture with OCR (Optical Character Recognition) for creating text-searchable PDF documents from scanned paper originals.
- The Adobe LiveCycle® suite of products, which provide core document services designed to align with your organization's services-oriented architecture and leverage open standards including J2EE and XML. They provide key functions such as document generation, document control and security, document collaboration, and process management for electronic forms. These products include Adobe LiveCycle Forms.
- The Adobe PDF Library SDK, available by license, allows third-party developers to support PDF within their own standalone applications. Developers can implement and develop PDF solutions in desktop environments as well as a wide range of server platforms. They can also take advantage of full compatibility with the latest PDF specification; full interoperability with Adobe products; support on a broad range of platforms; as well as high-performance, scalable, and optimized PDF processing.

Related Adobe technologies

- Adobe LiveCycle Reader Extensions provide advanced services and functionality for documents and forms accessed by Adobe Reader clients, extending users' capabilities for participating in form applications or workflows without requiring the full Acrobat product.
- Adobe LiveCycle Policy Server enables document authors and IT administrators to create usage policies that dynamically control who can view a PDF document, determine whether the recipient can modify, copy, print, or forward the document, and establish document expiration dates.
- Adobe LiveCycle Document Security provides digital signature and encryption capabilities in a server environment, enabling your organization to securely automate essential business processes.

What Is the Acrobat SDK?

The Acrobat Software Development Kit contains samples (example source code), header files (to compile the samples), type libraries, simple utilities, and documentation.

These tools help you design and develop the following projects:

- External applications that communicate with and control Acrobat and Adobe Reader.
- Scripts written in JavaScript that control Acrobat and Adobe Reader.
- Plug-ins that extend the functionality of Acrobat Professional, Acrobat Standard, and Adobe Reader.

Where can I get the Acrobat SDK?

To receive the full SDK, developer technical support, marketing assistance, and other benefits, join the ASN Developer Program.

The Acrobat SDK is available for free on the Adobe Solutions Network (ASN) web site.

To access the Acrobat SDK, see http://www.adobe.com/go/acrobat_developer.

User and developer resources

This section describes the user and developer resources available for the Acrobat product and SDK.

What do I need to download from the web to get the Acrobat SDK?

The Acrobat SDK is made up of many technical documents and samples for different operating systems. You can download the installer from the web site to obtain the entire Acrobat SDK, or you can download each piece separately using the “exploded” version. You will need to determine what you would like to do with the SDK, and then check if there is a sample that gets you started in the right direction.

See http://www.adobe.com/go/acrobat_developer to access the Acrobat SDK.

Where can I get help with Acrobat product issues?

For information and support for the Acrobat and LiveCycle products, you can visit the following sites:

Acrobat product information: http://www.adobe.com/products/acrobat/.

Acrobat product support: http://www.adobe.com/support/products/acrobat.html.

Adobe LiveCycle product information: http://www.adobe.com/products/server/.

For information on using or installing Adobe Acrobat products, start at the Acrobat Product Support Page, which contains free downloads of the latest Acrobat product updates, answers to top product support issues, a searchable support database, user forums, and product support telephone numbers.

For international customers, see http://www.adobe.com/supportservice/intlsupport.html.

Acrobat SDK development support resources

In general, Acrobat Developer Support addresses software development with the Acrobat core application programming interface (API), as documented in the Acrobat SDK. Non-ASN members can buy support cases, except in Japan. For Adobe Developer Support information, see http://www.adobe.com/go/acrobat_developer.

Acrobat Developer Support does not support use of the product that does not involve the Acrobat core API. Any non-programmatic issues, such as questions about installing, using, customizing, or deploying Acrobat, should be directed to Acrobat Technical Support rather than Adobe Developer Support. The supported development activities using the Acrobat SDK include those for which the product is designed, tested, and licensed.

Note: Each SDK version will continue to be supported for one major release after the next SDK version is released.

How can I contact Adobe for technical support issues?

If you have a technical support issue and do not find your answer online, you can contact Adobe Acrobat Technical Support at the telephone numbers listed on the Technical Support Phone Numbers Page: http://www.adobe.com/support/phonenumbers/main.html.

Technical support for the Acrobat SDK (“Acrobat Developer Support”) is included with an ASN membership. It is available to Adobe Solutions Network (ASN) Developer Program Premium members or may be purchased (except in Japan). For more information, see http://www.adobe.com/go/acrobat_developer.

Note: The UNIX version of the Acrobat SDK is available from http://www.adobe.com/go/acrobat_developer. It has release notes containing UNIX-specific information.

You will find answers to many common technical issues in the Acrobat Developer Knowledge Base at http://support.adobe.com/devsup/devsup.nsf/main.html.

What is the ASN Developer Program and how do I join?

The Adobe Solutions Network Developer Program includes marketing tools and technical tools. For complete information on the benefits of the ASN Developer Program, visit the Adobe Solutions Network page at http://www.adobe.com/go/acrobat_developer.

For developer information related to Adobe LiveCycle products, see http://www.adobe.com/devnet/livecycle/.

Contact information

To contact Adobe Systems, visit the ASN Developer Program contact page at http://www.adobe.com/go/acrobat_developer.

Technical support

Support for the Adobe Acrobat SDK is available to Adobe Solutions Network (ASN) Developer Program Premium members or may be purchased (except in Japan).

Technologies supported

The Acrobat core API; interapplication communication (IAC), including Apple events and DDE; PDF content generated by Adobe products; the pdfmark operator; command-line and initialization file interfaces; and installer customization.

Regular notification of updates and upgrades

Members receive e-mail regarding updates to technical notes, development tools, and development software.

Acrobat Software Development Kit (SDK)

See What Is the Acrobat SDK?

What Adobe resources are recommended to Acrobat developers?

For plug-ins and applications available from Adobe or third parties, Acrobat/PDF service providers, or links to pages on how to use the available plug-ins see http://www.adobe.com/go/acrobat_developer.

For white papers and case studies of what can be done and what has been done with Acrobat see http://www.adobe.com/products/acrobat/main.html.

What developer resources are provided by outside companies?

PDFZone (http://www.pdfzone.com/) provides resources that may be of help with developing and using the Acrobat products. These resources include forums, mailing lists, and a mailing list archive.

Licensing and distribution

This section provides information about Acrobat licensing and distribution, as well as technical and licensing limitations.

What should I know about Acrobat licensing and distribution?

Some of the Acrobat products may be licensed for commercial redistribution.

Products built around Acrobat and Adobe Reader must comply with the Acrobat End-User License Agreement (EULA). For example, customers who purchase third-party developer products that use Acrobat must still comply with the Acrobat EULA.

Adobe Reader

You may distribute Adobe Reader, subject to the terms set forth in the license agreement. However, you must use the installer that comes with Adobe Reader. You may invoke the Adobe Reader installer from your installer, but the Adobe Reader installer must remain intact and the End User License Agreement (EULA) must be displayed at first use. For information on customizing the Adobe Reader installer, see How can I customize the Acrobat installer?.

If you would like to distribute Adobe Reader, see the Adobe Reader distribution page given below in Additional resources.

Acrobat Capture (fee-based, license required)

With Acrobat Capture, you can submit an image file to a Capture workflow, where it will be “captured,” converted to PDF, and made available to the user. Capture can also convert paper to PDF and includes OCR to recognize text; it preserves the visual layout of the document. Capture is a stand-alone product that is licensed separately. It is not included as part of the Acrobat product. Low-volume Capture requires a hardware dongle that counts the pages processed; dongles are available with various page limits. For OEM’s and integrators, Adobe offers special pricing for Capture.

Acrobat toolkits

- Forms Data Format (FDF) Toolkit:

  Visit the Acrobat Family Developer Center at http://www.adobe.com/go/acrobat_developer to obtain the FDF Toolkit.

- Adobe PDF Library (fee-based, license required):

  The Adobe PDF Library is an object code library that can be linked to your application. The Adobe PDF Library is not free and is not available for free download from the Adobe public web site. However, the following URL provides information about the product: http://www.adobe.com/go/acrobat_developer.

Additional resources

Adobe Reader distribution: http://www.adobe.com/products/acrobat/distribute.html?readstep

Volume software licensing: http://www.adobe.com/aboutadobe/openoptions/main.html

What are the technical and licensing limitations

The following is a list of Acrobat SDK uses that are not supported. Most activities are identified as *technically infeasible* and/or *contrary to licensing*.

- Use of any Acrobat product in a multithreaded way (*technically infeasible*).

  Any multithreaded access to the Acrobat core API is likely to crash or hang the application. Acrobat makes the following methods available that can help a plug-in or application manage its thread's access to the Acrobat core API, as documented in the *Acrobat and PDF Library API Reference*:

  ```
  AVAppRegisterNotification
  AVAppRegisterIdleProc
  ```

  When registering for a notification, the method is passed a function to be called by the Acrobat viewer when the event occurs. Registering for an `IdleProc` calls the function when nothing else is occurring. All notifications and `IdleProcs` are queued and called in order. Registering for notifications of events or `IdleProcs` can help a multithreaded application ensure that Acrobat is not accessed by multiple threads simultaneously; however, it is still the application's responsibility to manage its threads that access Acrobat.

- Use of any Acrobat product as a server process accessed by multiple clients, unless it is specifically stated in product documentation as designed and licensed for such purpose (*technically infeasible* | *contrary to licensing*).

  Unless specifically stated in the product documentation and licensing agreement, Acrobat products are not designed, tested, or licensed for this purpose. This use is in violation of the licensing restrictions, as described in the End User License Agreement displayed during Acrobat installation and is not supported by Acrobat Developer Support.

- Use of Distiller as a Windows NT service (*technically infeasible*).

  Acrobat Distiller requires the ability to open a window on the desktop to run. It is not possible to use Distiller as a Windows NT service.

  Note: For customers needing this capability, the Acrobat 7.0 Professional product provides for *watched folders*.

- Use of `MenuItemExecute` to bring up Acrobat dialog boxes when a PDF file is displayed in an external window using OLE automation (*technically infeasible*).

  Due to problems of managing window focus, using the Acrobat dialog boxes (using `MenuItemExecute`) when a PDF file is displayed in an external window using OLE automation is not supported. The actions executed by the dialog box may or may not affect the intended PDF file and there can be problems of windows not redrawing properly. The only supported workaround is to use the OLE automation methods or to develop a plug-in to achieve any functionality not available in the OLE API to Acrobat. The Adobe Reader Integration Key License Agreement only permits displaying in an external window when Acrobat is used, not Adobe Reader.

- Use of the *PDF Reference* to develop a third-party application that writes PDF files without the use of Acrobat products.

  Note: The Adobe PDF Library, available under license, can be used to simplify development of these types of applications.

  We do not provide support to developers who are creating their own PDF generation capabilities without the use of Adobe products. The *PDF Reference* is the best resource for this kind of development. The use of Adobe products to create PDF files as a benchmark for your own development is recommended. Acrobat Developer Support will not debug PDF files created with non-Adobe products. Questions regarding the completeness or accuracy of the *PDF Reference* will be answered. The *PDF Reference* is available for free download from http://www.adobe.com/go/acrobat_developer.

- Interapplication communication (IAC) between a plug-in and a third-party application.

  Interapplication communication between a plug-in and a third-party application does not differ significantly from interapplication communication between two stand-alone applications. Documentation for your development platform's API and development environment are the best resources for this type of development. The Acrobat SDK contains two samples that can serve as examples.

  - `DDEServer` demonstrates DDE communication between a stand-alone application and a plug-in.
  - `ExternalWindow` uses Windows messaging to communicate with a stand-alone application.

  One typical difficulty for developers occurs when a multithreaded stand-alone application communicates with a plug-in. See Developing for Adobe Reader for more information.

- Use of platform API methods or the API methods of applications other than Acrobat.

  Acrobat Developer Support can help you with the API to the Acrobat family of products. Questions regarding the use of platform API methods should be directed to the manufacturer of your operating system.

- Use of the ActiveX® control or Netscape plug-in to display a PDF file in an external application besides Internet Explorer or Netscape. The methods used by Acrobat to display a PDF file in Netscape and Internet Explorer are intended only for use with these browsers. Use of the ActiveX Control and Netscape plug-in installed by Adobe Reader is not licensed to other applications. Development with these interfaces is not supported and no documentation is available.

- Automating the import of image files using the Import plug-in to Acrobat (*technically infeasible*).

  The Import plug-in to Acrobat does not provide an API that allows it to be called from a plug-in or another application. Executing the Import Image menu item with `MenuItemExecute` brings up a dialog box requiring user input.

  For information about other developer resources, see Developing for Adobe Reader.

What are some common uses of the Acrobat SDK?

The Acrobat SDK enables you to customize or modify how Acrobat looks and operates, and contains the information necessary to extend Acrobat's functionality to meet your needs. It documents and demonstrates the Acrobat core API capabilities, how to use the API, and how to use and debug JavaScript for Acrobat. Some common uses of the Acrobat SDK are described below. To understand the types of applications you can implement with the Acrobat SDK, see the *Overview* document. To see examples of the types of applications that can be created with the Acrobat SDK, including ones specifically intended for Windows and Mac OS platforms, see the *Guide to SDK Samples*.

PDF file viewing

You can view a PDF file in your own application's window using Acrobat. For guidelines on how this may be done, see How can I display a PDF file in an external application window? You can also display a PDF file as an ActiveX document in applications using simplified browser controls, by using an automation interface available in both Adobe Reader and Acrobat. For details, see *Developing Applications Using Interapplication Communication* and the *Interapplication Communication API Reference*.

PDF file printing

You can customize how various portions and layers of a PDF file are printed using JavaScript for Acrobat, interapplication communication (IAC), or plug-ins. For more information, see the *Overview*.

Customizing the user interface

You can customize the user interface for specific workflows by writing a plug-in or JavaScript to add or change menu bars, menu items, or tool buttons that appear in the user interface. For more information, see the *Overview*.

Supporting workflows

You can support custom workflows for PDF documents in the following ways:

- Adding elements such as watermarks, bookmarks, and links
- Creating custom templates
- Automatically filling form fields
- Adding metadata
- Setting up custom searches and building catalogs
- Managing page orientation and fonts
- Setting up batch jobs
- Automating PDF document creation through control of Distiller
- Merging or splitting documents
- Directly accessing the file system to save and open PDF files to and from directory structures
- Saving in various formats such as XML
- Optimizing performance of PDF viewing
- Applying stamps
- Using security settings
- Exchanging information with databases
- Collaboration servers or web services
- Manipulating engineering drawings

For more information, see the *Overview*.

Controlling an Acrobat or Adobe Reader session

You can remotely control an Acrobat Professional, Acrobat Standard, or Adobe Reader session. See the `DDEServer` sample plug-in for an example of how to communicate with an Acrobat plug-in from an external application using DDE, and the `RemoteControlAcrobat` interapplication communication sample. For more information, see the *Guide to SDK Samples*.

How can I customize the Acrobat installer?

Adobe provides various ways in which you can deploy Acrobat to a large number of systems.

You can find documentation about enterprise installation, including information about tools for customization and installation as well as guidelines for extending enterprise applications, at: http://www.adobe.com/products/acrobat/planning.html.

For technical notes about how to deploy Acrobat and Adobe Reader throughout your entire enterprise, see http://www.adobe.com/go/acrobat_developer.

Solutions for enterprise deployment are provided at http://www.adobe.com/products/acrobat/deployment.html, where you will find a variety of technical information on enterprise deployment, including the following:

Citrix User Document — An explanation of how to use the Citrix Independent Computing Architecture (ICA) protocol (via the Citrix MetaFrame 1.8 add-on) with Acrobat.

SMS User Document — Information on deploying Acrobat to the enterprise via SMS.

WTS User Document — Information on deploying Acrobat to the enterprise via Windows Terminal Service.

InstallShield Tuner for Acrobat — Enterprise installation support for Acrobat.

Tivoli User Document — Support for enterprise deployment of Acrobat Professional, Acrobat Standard, Adobe Reader, and Acrobat Elements using Tivoli.

2 Understanding the Acrobat SDK

In this chapter, various aspects and requirements of forms, the use of plug-ins and JavaScript, development issues related to Acrobat and Reader, PDF documents, and enterprise installation are discussed.

About forms

This section provides answers to questions about using forms in Acrobat and Adobe Reader.

What are the requirements for using Acrobat forms?

You can create and fill in Acrobat forms in PDF files, and submit or import form data. These forms are typically used for viewing or printing information, filling in information, selecting options, digitally signing documents, and exchanging information with databases. Note that users of Adobe Reader can fill in and submit forms, and can save the form data to disk if they have LiveCycle Reader Extensions, and Acrobat 8 Professional can enable a form to be saved locally with Adobe Reader 7.0 and later. Once a user fills out an Acrobat form, the data can be submitted to a server for processing. Data can be exported from a PDF form into Forms Data Format (FDF) or XML-based FDF (XFDF). Data can be imported into a PDF form if it is in FDF, XFDF, XML, or TXT format. Acrobat forms support the following formats: FDF, XFDF, tab-delimited text, and XML.

Every FDF file contains a reference to a PDF file for which the data is intended, designated with the `/F` key inside the FDF file (unless the FDF file is for the same PDF from which you submitted your data). When Acrobat or Adobe Reader receives an FDF file, it opens the corresponding PDF file, and fills the form fields with the data from the FDF file. If the PDF file is referenced by a URL (for example, http://example.com/file.pdf), the FDF file must be sent to the server in response to a `submit` action from a PDF form. For more information on the FDF format, which is based on PDF, see the *PDF Reference*.

Acrobat plug-ins can programmatically import FDF data into a PDF file from a local file system using the HFT made available by the Forms plug-in. OLE automation can be used to programmatically add, modify, or delete form fields, import or export FDF data, execute scripts written in JavaScript, and much more. For more information, see the *Acrobat and PDF Library API Reference*.

What are the requirements for using XML forms?

The XML Forms Architecture (XFA) supports the production of XML-based business form documents known as dynamic XML forms. Dynamic XML forms are suited for exchanging information with web services and enterprise infrastructures. Dynamic XML forms include the following features:

- Fields that may be moved and dynamically resized even to the point of spanning multiple pages
- Repeating subforms that can be spawned as needed with page contents shifting accordingly
- Page elements that can be shown or hidden as needed

Use LiveCycle Designer to create or edit templates. For more information, see http://www.adobe.com/products/server/.

Dynamic XML forms support both the XML and XDP formats. For more information, see *Developing Acrobat Applications Using JavaScript* and the *JavaScript for Acrobat API Reference*.

What is the FDF Toolkit?

The FDF (Forms Data Format) Toolkit is a thread-safe API for writing a server application that generates FDF data or parses FDF data from a form created by the Acrobat Forms plug-in. If you need to parse FDF files submitted from a PDF form, or generate FDF files to be submitted to a PDF form, you can use the FDF Toolkit. The FDF Toolkit supports COM on Windows, C and Perl on Windows, Solaris™, AIX® or Linux®, and Java VMs compatible with versions 1.2 or later.

For detailed information, see the documentation included with the FDF Toolkit.

Plug-ins and JavaScript

This section covers questions about the use of plug-ins and scripts.

What is a plug-in?

Plug-ins are dynamically-linked extensions to Acrobat or Adobe Reader that enhance or customize its functionality. They can appear in the user interface in a variety of ways, and can respond to events that occur in the application.

A plug-in is a program written in ANSI C/C++ that uses the Acrobat public API. Its purpose is to add functionality to Acrobat Professional, Acrobat Standard, or Adobe Reader. A plug-in program file must be stored in the Plug_ins folder in the user's Acrobat or Adobe Reader installation, and its functionality becomes available when the Acrobat or Adobe Reader application is started.

There are three types of plug-ins:

Regular plug-ins: The samples in the SDK are of this type.

Reader-enabled plug-ins: These are plug-ins that are developed with permission from Adobe, and require special processing to load under Adobe Reader. For more information, see *Developing for Adobe Reader*.

Certified plug-ins: For more information, see What is a certified plug-in?

Plug-ins are deployed differently on each platform:

- DLLs on Windows. Note, however, that plug-in names must end in .API, not .DLL.
- Code fragments on Mac OS.
- Shared libraries on UNIX®.

What is a certified plug-in?

A certified plug-in is considered the most secure type of plug-in available for Acrobat or Adobe Reader. Certified plug-ins are only available from Adobe, and are tested extensively to ensure that they are in full compliance with the Acrobat security model. The Acrobat and Adobe Reader user interfaces provide a checkbox that may be used to ensure that only certified plug-ins load. In Acrobat or Adobe Reader, select Edit > Preferences > Startup and check Use Only Certified Plug-ins.

Note: There is currently no way for third-party plug-ins to be certified by Adobe.

How does a script written in JavaScript differ from a plug-in?

JavaScript for Acrobat was originally developed for Acrobat forms, and has been greatly expanded. You can use JavaScript for Acrobat to carry out the following tasks:

- Develop and process PDF forms
- Customize the behavior and appearance of the PDF document
- Facilitate online team review
- Implement security policies
- Interact with databases and web services
- Customize the Acrobat application itself through access to its user interface components

While developers writing plug-ins have direct access to the Acrobat Core API, JavaScript applications tend to be easier to write and implement, since they are developed using the editor and debugger that are included in Acrobat Professional. JavaScript applications are also easier to distribute since they can be included directly within a PDF file, whereas plug-ins must be placed in the Plug_ins folder by either an installer or the user. JavaScript for Acrobat can be used across multiple platforms, while a plug-in must have separate versions for each platform in order to handle certain platform-specific issues.

It is also worth noting the following comparisons between JavaScript for Acrobat and plug-ins:

- Since it is interpreted rather than compiled, execution of JavaScript for Acrobat code is slower than plug-in code. However, the difference tends to be noticeable only in computationally intensive applications, such as a full text search in a large PDF file.
- JavaScript for Acrobat cannot be used to add content to a PDF file via a content stream.
- JavaScript for Acrobat is well-suited to quickly adding user interface capabilities.

Finally, while there is a certain amount of overlap between what can be done in JavaScript and what can be done in a plug-in, some capabilities are only available in JavaScript while others are only available with a plug-in, as summarized in the table below. For more information, see the *Overview*.

Capability	**JavaScript**	**Plug-in**
Use SOAP and web services	Yes	No
Play multimedia	Yes	No
Automate email review workflow	Yes	No
Search Acrobat Help	Yes	No
Use Acrobat security policies	Yes	No
Use content stream and other low-layer access	No	Yes
Create new menus or toolbars	No	Yes
Create new annotation or action types	No	Yes
Modify the scaling factor for large PDF file sizes	No	Yes
Access platform-specific services or events	No	Yes
Access wireframe drawing mode	No	Yes

Developing for Adobe Reader

This section addresses specific issues when developing plug-ins for Adobe Reader, and highlights the API differences between Acrobat and Reader.

Where can I find guidelines on developing Adobe Reader plug-ins?

If you are considering plug-ins for Adobe Reader, bear in mind the following:

- You may not develop an Adobe Reader plug-in without approval from Adobe. There is a web-based application where you describe your plug-in and submit the information to Adobe; Adobe will then review it and let you know whether your application has been approved. The application and the Adobe Reader Integration Key Licensing Agreement can only be submitted as a web form and are found at http://www.adobe.com/go/acrobat_developer.
- There is a fee to obtain the enabling key.
- There is a restricted set of APIs available in Adobe Reader. For technical limitations, see the following documents:
  - *Developing Plug-ins and Applications*
  - *Acrobat and PDF Library API Reference*
  - *Developing Applications Using Interapplication Communication*
- For information on how to access the Host Function Tables (HFTs) available to Adobe Reader plug-ins, see *Developing Plug-ins and Applications*.
- For instructions on how to enable your plug-in for Adobe Reader, see *Developing Plug-ins and Applications*.

What are the API differences between Acrobat and Adobe Reader?

Acrobat provides a “full-featured” development environment that includes the entire Acrobat core API. There are some small differences between the public APIs available in Acrobat Professional and Acrobat Standard. These are documented in *Developing Plug-ins and Applications* and the *Acrobat and PDF Library API Reference*.

The APIs that may be used for Adobe Reader are limited technically and legally. Technical limitations are documented in *Developing Plug-ins and Applications* and *Developing Applications Using Interapplication Communication*.

Both Acrobat and Adobe Reader accept plug-ins. The primary difference between the two is that, in general, Adobe Reader can neither make changes to a file nor save a file. API methods that change a file in such a way that a save would be required are not available in Adobe Reader.

Developing for Adobe Reader

Adobe Reader only accepts “Reader-enabled” plug-ins, which can only access a limited set of APIs. For information on enabling plug-ins for Adobe Reader, see *Developing Plug-ins and Applications* and *Developing for Adobe Reader*. Certain APIs are available to a Reader-enabled plug-in if the PDF document has been assigned additional usage rights, which is possible using LiveCycle Reader Extensions. For more information on this, see the *Overview*.

For additional legal restrictions, see the Adobe Reader Integration Key License Agreement.

What is a rights-enabled PDF document?

With a rights-enabled PDF document, the free and ubiquitous Adobe Reader c an be used to download, save, fill in, digitally sign, and submit electronic PDF forms. PDF documents are “enabled” using LiveCycle Reader Extensions or Acrobat 8.0 Professional. With a rights-enabled PDF document, developers are able to use the digital signature APIs for signing and signature verification. One major advantage of LiveCycle Reader Extensions is that it supports data automation through XML-based representation and transfer using SOAP, thus ensuring seamless integration with business logic and advanced data transport capabilities available within enterprise applications. For more information, see the *Overview*.

PDF documents

This section provides answers to questions surrounding the display and manipulation of PDF documents in the Acrobat SDK.

What ActiveX solutions are provided by the Acrobat SDK?

The PDF browser control automation object, described in *Developing Applications Using Interapplication Communication* and the *Interapplication Communication API Reference*, supports the loading and display of PDF documents as ActiveX documents, and its interface is available in both Acrobat and Adobe Reader.

Visual Basic .NET and Visual C# .NET

This section describes how Visual Basic .NET and Visual C# .NET are used in the interapplication communication support available in the Acrobat SDK.

How can I use Visual Basic .NET or Visual C# .NET with the Acrobat SDK?

The Windows version of Acrobat is an OLE automation server. In order to use the OLE objects made available by Acrobat, you must have the full Acrobat product installed on your system and the `Acrobat.tlb` file included in the project references for your Visual Basic project. This allows you to use the Visual Studio 2005 object browser to browse the OLE objects.

For Acrobat versions 5.0 and later, it is possible to access the JavaScript object model in Acrobat through the OLE automation `JSObject` interface. See the `JSObjectAccess` sample. For more information, see *Developing Applications Using Interapplication Communication*.

What are the best resources for getting started using Visual Basic .NET or Visual C# .NET

Developing Applications Using Interapplication Communication and the *Interapplication Communication API Reference* describe the objects and methods available in these languages, as well as guidelines for usage. These documents (as well as the API) were designed with C programming in mind, and programming with the API requires some familiarity with C concepts such as `enum`.

Besides the object browser, the best resources for programming in these languages are the sample projects. The samples demonstrate use of the Acrobat OLE objects and contain comments describing the parameters for the more complicated methods. For more information, see the *Guide to SDK Samples*.

The `Iac.bas` file in the InterAppCommunicationSupport\Headers folder contains the Visual Basic counterparts to the C `#define` statements and enumerations passed to many of the API methods.

What API methods are available to modify PDF documents?

Modifying the page contents of a PDF file is primarily accomplished using the Acrobat core API.

Using the Acrobat core API greatly simplifies modifying and creating PDF page contents. To demonstrate this, there are several “snippets” available from the `SnippetRunner` sample plug-in that show you how to add data to the contents of a page, while ensuring that the PDF file is still readable after modification. To attempt to do this without using the core API would be significantly more difficult and could result in an unreadable PDF file.

Acrobat’s automation interfaces are limited mainly to what a user can do through the user interface and cannot modify the contents of a page.

For more information on the Acrobat core API, see *Developing Plug-ins and Applications* and the *Acrobat and PDF Library API Reference*.

What options do I have to modify PDF documents without a C programming background?

You can make some modifications to PDF files through JavaScript for Acrobat. *Developing Acrobat Applications Using JavaScript* and the *JavaScript for Acrobat API Reference*, freely available for download from the Adobe web site, provide all the information you need to get started using JavaScript. They serve as a reference to the JavaScript objects, properties, and methods in Acrobat.

How can I extract text from PDF documents using the Acrobat SDK?

You can extract text with the Acrobat SDK in two ways:

- Use the Acrobat core API
- Use the Acrobat automation API

Through the core API, you can extract ASCII text from a PDF file using Acrobat and a plug-in developed in C or C++. The `TextExtraction` and `WordFinder` sample plug-ins demonstrate text extraction and can be used as starting points for your own plug-in. `AVConversion` methods also can be used to save PDF as text or rich text. In addition, the `SDKTextExtraction` sample in the first level of the SaveAsXML directory provides a good starting point for creating richer extraction tables. For more information, see the *Guide to SDK Samples*.

How can I display a PDF file in an external application window?

There are several ways to have the Acrobat program display a PDF file in an external application’s window. Acrobat must be installed on the system to view a PDF file in your own application’s window.

There is no best way that we suggest to display PDF files in your application; you should examine the following list for the most appropriate method for your situation. For more information, see *Developing Applications Using Interapplication Communication* and the *Interapplication Communication API Reference*.

Using Acrobat to view a PDF file in your own application’s window

Windows only:

- OLE automation, using the `OpenInWindowEx` command. This displays a live view of the PDF file in the OLE application window.

  For samples, see the Visual Basic and C++ `ActiveView` applications.

- OLE automation, using the `DrawEx` command. This displays a bitmap of the current page in the OLE application window.

  For samples, see the Visual Basic and C++ `StaticView` applications.

- Copy to clipboard. This copies a PDF image to the clipboard without requiring an `hWnd` or `hDC` using OLE automation.
- If you are using simplified browser controls in your application, you may treat the PDF document as an ActiveX document by using the `AcroPDF` object's `LoadFile` method. This automation interface is available in both Acrobat and Adobe Reader.

Mac OS only:

- The `AcroAppleEvents` sample application demonstrates rendering a PDF file into another application's window (see `DrawIntoWindowCommand` for details).

Windows and Mac OS:

- The `ExternalWindow` sample plug-in demonstrates a live view of the PDF file in a window created by the plug-in. You can extend this to display PDF files in an external application window.

Are multibyte font PDF documents supported by the Acrobat SDK?

Acrobat allows the creation, modification, and use of PDF files containing multibyte fonts. For more information, see the *Overview*.

How are security and encryption provided in PDF documents?

Adobe provides password-based and certificate-based encryption schemes with Acrobat. In the password-based scheme, Acrobat queries the user for a password before a file is opened. The authorization procedure to open a file or to set an owner password can be modified by creating a custom security handler. For more information on security handlers and encryption in PDF documents, see the *Overview*.

Full-text search

This section describes the tools and support available in the Acrobat Search plug-in.

What tools are available with Acrobat for full-text search?

Adobe provides a full-text search system, known as the Acrobat Search Plug-in, based on third-party technology. However, this does not preclude other search systems from integrating with Acrobat. The Acrobat search system was created using only public API and IAC calls, and you can easily remove it from Acrobat and replace it with another search technology. For more information, see the *Interapplication Communication API Reference*.

The Acrobat interface can be extended with new menus and toolbar icons to allow tight integration with your search plug-in and Acrobat. For example, buttons to invoke a search and to find the next or previous occurrences can be added to the toolbar.

For samples of adding menu items and toolbar buttons, see the related snippets in the `SnippetRunner` sample plug-in.

What tools are available for extracting and highlighting text?

For indexing and searching PDF files directly, Acrobat provides support through IAC and plug-in calls. For indexing PDF files, Acrobat provides text extraction APIs. For further information on extracting text from PDF, see How can I extract text from PDF documents using the Acrobat SDK?

How do I communicate with the Acrobat Search plug-in?

You can communicate with the Search plug-in via its plug-in API or via IAC (DDE or Apple events). Using either of these methods, you can control the Acrobat Search plug-in in the following ways:

- Control the list of indexes to search

  The search interface allows for searches to be performed on one or more of the available indexes. You can control the list of active indexes.

- Initiate a search with options

  You can pass query expressions and search settings to the Acrobat search plug-in and initiate the search, and the results will be presented to the user.

For an example, see the `SearchPDF` Visual Basic application. For documentation, see the *Acrobat and PDF Library API Reference*.

How do I create custom DocInfo fields for searching?

The Catalog feature is used to create a search index; its API is available to third parties via the Catalog plug-in provided with the Acrobat SDK. See the *Acrobat and PDF Library API Reference* and the *Interapplication Communication API Reference* for more information.

Once you have built the search index, you must create the custom `DocInfo` fields, which can be used to search document metadata. You can accomplish this task with either the Acrobat core API or using interapplication communication. See the *Acrobat and PDF Library API Reference* for more information.

At this point you may submit custom queries to the Acrobat Search plug-in to search by the custom fields, as demonstrated by the `SearchQuery` sample plug-in. You can include your custom `DocInfo` fields in the search by specifying them in the `sortSpec` parameter of the `SearchExecuteQuery` method.

How do I use the Windows command line with Acrobat and Adobe Reader?

You can display and print a PDF file with Acrobat and Adobe Reader from the command line. These commands are unsupported, but have worked for some developers. There is no documentation for these commands other than what is listed below.

Note: All examples below use Adobe Reader, but apply to Acrobat as well. If you are using Acrobat, substitute `Acrobat.exe` in place of `AcroRd32.exe` on the command line.

`AcroRd32.exe` ***`pathname`*** — Start Adobe Reader and display the file. The full path must be provided.

This command can accept the following options.

Option	Meaning
`/n`	Start a separate instance of Acrobat or Adobe Reader, even if one is currently open.
`/s`	Suppress the splash screen.
`/o`	Suppress the open file dialog box.
`/h`	Start Acrobat or Adobe Reader in a minimized window.

`AcroRd32.exe /p` ***`pathname`*** — Start Adobe Reader and display the Print dialog box.

`AcroRd32.exe /t path` ***`"printername" "drivername" "portname"`*** — Start Adobe Reader and print a file while suppressing the Print dialog box. The path must be fully specified.

The four parameters of the `/t` option evaluate to `path`, `printername`, `drivername`, and `portname` (all strings).

`printername` — The name of your printer.

`drivername` — Your printer driver's name, as it appears in your printer's properties.

`portname` — The printer's port. `portname` cannot contain any "/" characters; if it does, output is routed to the default port for that printer.