



Preface

Introduction

Welcome to the Adobe® *Acrobat® JavaScript Scripting Guide*. This scripting guide is designed to provide you with an overview of how you can use Acrobat JavaScript to develop and enhance standard workflows, such as:

- Printing and viewing
- Spell-checking
- Stamping and watermarking
- Managing document security and rights
- Accessing metadata
- Facilitating online collaboration
- Creating interactive forms
- Customizing interaction with Web Services
- Interacting with databases

Here you will find detailed information and examples of what the Acrobat JavaScript capabilities are and how to access them, as well as descriptions of the usage of the SDK tools. Acrobat JavaScript is a powerful means by which you can enhance and extend both Acrobat and PDF document functionality.

What is Acrobat JavaScript?

Acrobat JavaScript is a language based on the core of JavaScript version 1.5 of ISO-16262, formerly known as ECMAScript, an object-oriented scripting language developed by Netscape Communications. JavaScript was created to offload Web page processing from a server onto a client in Web-based applications. Acrobat JavaScript implements extensions, in the form of new objects and their accompanying methods and properties, to the JavaScript language. These Acrobat-specific objects enable a developer to manage document security, communicate with a database, handle file attachments, manipulate a PDF file so that it behaves as an interactive, web-enabled form, and so on. Because the Acrobat-specific objects are added on top of core JavaScript, you still have access to its standard classes, including **Math**, **String**, **Date**, **Array**, and **RegExp**.

PDF documents have great versatility since they can be displayed both within the Acrobat software as well as a Web browser. Therefore, it is important to be aware of the differences between Acrobat JavaScript and JavaScript used in a Web browser, also known as *HTML JavaScript*:

- Acrobat JavaScript does not have access to objects within an HTML page. Similarly, HTML JavaScript cannot access objects within a PDF file.
- HTML JavaScript is able to manipulate such objects as **Window**. Acrobat JavaScript cannot access this particular object but it can manipulate PDF-specific objects.

If you have used previous versions of Acrobat JavaScript, please note that there are a number of new JavaScript features and enhancements in version 7:

- JavaScript Byte Code
- Adobe PDF document creation
- Additional usage rights
- Engineering features
- File attachments
- Additional language support
- Forms authoring and management
- Review, markup, and approval
- Document security and digital signatures
- Accessibility
- Print production
- XML capabilities

Audience

It is assumed that you are an Acrobat solution provider or power user, and that you possess basic competency with JavaScript. If you would also like to take full advantage of Acrobat's web-based features, you will find it useful to understand XML, XSLT, SOAP, and Web services. Finally, if you would like to utilize Acrobat's database capabilities, you will need a basic understanding of SQL.

Purpose and Scope

The purpose of this guide is to:

- Describe how you can use the Acrobat JavaScript language as the primary vehicle with which to develop and deploy Acrobat workflow solutions.
- Provide you with easily understood, detailed information and examples of the Acrobat JavaScript scripting features.
- Provide you with references to other resources where you can learn more about Acrobat JavaScript and related technologies.

After reading this guide and completing the exercises, you should be equipped to start using Acrobat JavaScript. During the development process, you will find that the descriptions and examples in this guide will provide you with enough direction and background to enable you to successfully complete your projects.

Assumptions

This guide assumes that you are familiar with the non-scripting elements of the Acrobat 7 user interface that are described in Acrobat's accompanying online help documentation. To work through the exercises in this guide, you will need to use Acrobat 7 Professional.

How To Use This Guide

This guide includes exercises that give you an opportunity to work directly with Acrobat JavaScript. If you plan to practice any of the exercises, do the following:

1. Be sure that you have Acrobat Professional installed on your Windows® or Macintosh® workstation. The exercises are designed to work on Windows and Macintosh versions of Acrobat, unless otherwise noted.
2. Create and use the same directory on your machine for all the JavaScript exercises you would like to try. You will use this directory to store the PDF documents and other files used in the exercises.
3. The Acrobat SDK contains a set of `.zip` files you will need to work through the exercises. You should extract the contents of these files to your local directory.

Font Conventions Used in This Book

The Acrobat documentation uses text styles according to the following conventions.

Font	Used for	Examples
monospaced	Paths and filenames	C:\templates\mytmpl.fm
	Code examples set off from plain text	These are variable declarations: AVMenu commandMenu,helpMenu;
monospaced bold	Code items within plain text	The GetExtensionID method ...
	Parameter names and literal values in reference documents	The enumeration terminates if proc returns false .
monospaced italic	Pseudocode	ACCB1 void ACCB2 ExeProc(void) { <i>do something</i> }
	Placeholders in code examples	AFSimple_Calculate(<i>cFunction</i> , <i>cFields</i>)
blue	Live links to Web pages	The Adobe Solutions Network URL is: http://partners.adobe.com/asn/
	Live links to sections within this document	See Using the SDK .
	Live links to code items within this document	Test whether an ASA tom exists.
bold	PostScript language and PDF operators, keywords, dictionary key names	The setpagedevice operator
	User interface names	The File menu
italic	Document titles that are not live links	<i>Acrobat Core API Overview</i>
	New terms	<i>User space</i> specifies coordinates for...
	PostScript variables	<i>filename</i> deletefile

Related Documents

This guide refers to the following sources for additional information about Acrobat JavaScript and related technologies:

- *Acrobat JavaScript Scripting Reference*
This document is the companion reference to this scripting guide. It provides detailed descriptions of all the Acrobat JavaScript objects.
- *Adobe Acrobat Help*
This online document is included with Acrobat.
- Acrobat Solutions Network
<http://partners.adobe.com/asn/>
- *PDF Reference*
- *Guide to SDK Samples*
- *Developing for Adobe Reader*

If, for some reason, you did not install the entire Acrobat SDK and you do not have all the documents, please visit the Adobe Solutions Network Web site to find the documents you need.