# Samples

## IAC

UNIX viewers do not support IAC.

### 1 AcroAuto

The AcroAuto folder contains several simple samples which demonstrate how to use OLE automation using a C-like interface. The ctest and launch samples illustrate simple examples; test is a bit more complex. These samples link in the acroauto.lib file at link time. The acroauto.dll and acroauto.h make all of the OLE automation functions exported by the viewer available as C calls.

#### Illustrates

- Using OLE automation from a C program

#### Source

- IAC\WIN\SAMPLES\AcroAuto

### 2 AEView

The AEView sample demonstrates controlling various components of Exchange for the Macintosh® using Interapplication Communication and C programming. Among the techniques shown are the following:

- making Exchange render a PDF file into another application's window (see the DrawIntoWindowCommand() function for details)
- custom interface to the Search plug-in
- inhibiting PDFWriter printing dialogs
- creating PDF files dynamically with the Distiller

#### Source

- IAC:MAC:SAMPLES:AEView

### 3 CAPTDDE

A 32-bit sample demonstrating DDE communication with Capture.

#### Source

- IAC\WIN\SAMPLES\MFC\CAPTDDE

# 4 DDEOpen

Demonstrates a basic DDE connection to the Acrobat viewer. It does a DocOpen on the file test.pdf on the c: drive.

### Source

- IAC\WIN\SAMPLES\MFC\DDEOPEN

# 5 Distill

Demonstrates communication with the Distiller program.

### Source

- IAC\WIN\SAMPLES\MFC\DISTILL

# 6 Draw and VBDraw

These examples illustrate how to render the PDF page contents into another application's window using OLE 2.0. VBDraw does this via Visual Basic, Draw with Visual C++.

### Caveats

VBDraw requires these included files in your windows\system directory in order to work (if you don't have Visual Basic installed):

- cmdialog.vbx
- vb0a300.dll
- vbdb300.dll
- vbrun300.dll

The Draw example requires Visual C++ 1.5 to be installed in order to compile this example. If OLE is not working, these examples will not work.

### Source

- IAC\WIN\SAMPLES\MFC\DRAW
- IAC\WIN\SAMPLES\VB\VBDraw

# 7 Notes

Demonstrates how to use the Lotus Notes APIs and the Acrobat OLE Automation APIs to manipulate PDF files stored in Notes databases.

This example works with a Notes template to build a Notes database for reviewing documents. The documents are PDF files embedded into Notes using OLE. The Notes template will manage the review process so that as documents are reviewed in Acrobat by adding annotations they will become responses to the original document. After a number of reviewers have finished reviewing the document the annotations from all the reviewers can be collated into a single document by using the Acrobat and Notes APIs.

### Illustrates

Integrating Acrobat with Lotus Notes APIs. In particular this example demonstrates how to launch an Acrobat document embedded with OLE in a Notes database. Once the Acrobat document is launched OLE automation is used to manipulate the annotations in a PDF file.

### Source

- IAC\WIN\SAMPLES\MFC\Notes

## 8 OIW and VBOIW

These examples illustrate how to display the AVDoc window of a PDF file within another application's window using OLE 2.0. VBOIW does this via Visual Basic, OIW with Visual C++. The examples also demonstrates controlling the Distiller and PDFWriter. Note that there is a facility for doing this with AppleEvents.

### Caveats

VBOIW requires these included files in your windows\system directory in order to work (if you don't have Visual Basic installed):

- cmdialog.vbx
- vb0a300.dll
- vbdb300.dll
- vbrun300.dll

The OIW example requires Visual C++ 1.5 to be installed in order to compile this example.

When using the OpenInWindow OLE command, developers should keep in mind that the viewer does not enable all of the menu items, making most standard menu functions inoperable. One alternative would be to use the PD level OLE call Draw, which is demonstrated in the DRAW example.

In addition cross document links will not work when using OpenInWindow.

### Source

- IAC\WIN\SAMPLES\MFC\OIW
- IAC\WIN\SAMPLES\VB\VBOIW

## 9 OpenAll—Distiller

Openall—Distiller demonstrates how to remotely control the Acrobat Distiller using AppleScript.

### Source

- IAC:MAC:SAMPLES:OpenAll—Dist

## 10 Pdfwctrl

This sample demonstrates using the Windows® API Escape function with PDFWriter.

### Source

- IAC\WIN\SAMPLES\MFC\PDFWCTRL

## 11 PrintPages

This Apple script prompts the user for a PDF file name and prints the file.

### Source

- IAC:MAC:SAMPLES:Scripts:PrintPages.txt

## 12 Rotate

Rotate is an AppleScript example which shows how to drive the Acrobat viewer using Apple events. It provides the following functionality:

- querying the user to select a directory containing PDF files
- obtaining a list of files in that directory
- checking if the creator type for a file is correct for a PDF file
- rotating all of the pages of an opened PDF file an arbitrary amount
- closing and saving the changes made
- demonstrates basic AppleScript commands to put up dialog boxes querying for results
- extracting a file path from a folder of files

### Source

- IAC:MAC:SAMPLES:Scripts:Rotate.txt

## 13 Select Text

This AppleScript example demonstrates how to launch the viewer, query the user to open a PDF file, and selection of text within the PDF document.

### Source

- IAC:MAC:SAMPLES:Scripts:Select Text.txt

## 14 Set

This AppleScript example demonstrates the launching of the viewer, querying the user to open two different PDF files, and then setting various parameters including turning the splash screen on and off, setting the active document, modifying the zoom factor and type, and changing the page

number that is being displayed. It also demonstrates how to set an active tool and how to quit the application without saving changes

### Source

- IAC:MAC:SAMPLES:Scripts:Set.txt

## 15 VBDraw

See Section 6 on page 2.

## 16 VBOIW

See Section 8 on page 3.

## 17 VBSrch and Srchcomm

VBSrch is a Visual Basic front-end program that demonstrates how to communicate with Acrobat Search using functions exported by a dll called srchcomm. The dll provides an interface to specify the language type, sort order, number of words returned, and word options. This is done by making calls to the srchcomm dll in order to fill out a search query, which is then sent to Acrobat Search. The srchcomm dll can add, remove, enable or disable indexes.

This dll can also be used in an Visual C++ application or in an Acrobat plug-in.

### Source

- IAC\WIN\SAMPLES\VB\VBSRCH
- IAC\WIN\SAMPLES\VB\VBSRCH\SRCHCOMM

# Plug-Ins

The following plug-ins have been built and their source included in the Acrobat SDK so that developers can try them out.

### Synopsis of supported plug-ins and viewer mechanism that plug-in illustrates:

- acrobat2.1howto or acro21— demonstrate Acrobat 2.1 API
- addps — modifies the print stream
- balloon — custom action/annotation/import HFT/private data storage
- debuggerwindow — export HFT
- hftquery — communication with another plug-in through the HFT
- imagesel — selection server
- modpdf — append to the contents stream of a page
- notify — register for notifications
- openall — conversion of non-PDF files into PDF files using Distiller and PDFWriter
- pistarter — project setup for different compilers
- progbar — custom progress monitor
- rfs — replacement file system
- rplcdemo — method replacement
- setsec — file security
- snapzoom — coordinate system manipulation
- stamper — custom tool/annotation handler
- subcrypt — security handler
- template — style example/starter plug-in
- textfltr — window handler
- wordfind — text extraction

## 18 Acrobat2.1HowTo; Acro21

This plug-in shows how to use the Acrobat 2.1 API. It demonstrates which files to modify in order to build Acrobat 2.1 plug-ins. The sample invokes the PDDocClearFlags() method when the tool button is clicked.

### Source

- PLUGINS:MAC:SAMPLES:Acrobat 2.1 How
- PLUGINS\WIN\SAMPLES\ACRO21

# 19 Acrobat Toolkit

These directories in the Acrobat Plug-Ins SDK contain the Acrobat Toolkit, a Unix library that extracts text from PDF files. Versions for Windows NT (Win32), IRIX™, HP-UX, AIX, SunOS™, and Solaris® are provided. This version of the toolkit produces words that are consistent with the shipping Acrobat 2.0 products for Macintosh and Windows.

This ToolKit is no longer free; if you would like to redistribute this library, contact the Adobe Developers Association for a license agreement.

All methods except PDFTKinit are documented in Technical Note #5168, *Acrobat Viewer Plug-In API On-line*, available in the Acrobat Plug-Ins SDK. PDFTKinit initializes the toolkit. It must be called before any of the other toolkit calls.

For more information see PDFTK/ReadMe.

### Illustrates

- PDWordFinder

### Source

- PRODUCTS/ACROTK/AIX
- PRODUCTS/ACROTK/HPUX
- PRODUCTS/ACROTK/HPUX
- PRODUCTS/ACROTK/SGI
- PRODUCTS/ACROTK/SUNOS

# 20 AddPS

This plug-in modifies the viewer’s print stream. It adds an AddPS menu item to the Tools menu. When this is selected, the user gets a standard file selection dialog box. The file selected will be opened and its contents sent to the print stream at the start or end of each page of the PDF file being printed. It’s assumed that the user will select a PostScript file in order for this to work. The PostScript file can contain anything, but it’s up to you to make sure things still print. To turn off this behavior, choose the AddPS command and cancel the file selection dialog. The program is set up to add the PostScript file to the beginning of each page, but can be conditionally compiled to add it to the end of each page.

The example PostScript file Draft.ps causes the word DRAFT to be printed at 45 degrees in the middle of the page.

### Illustrates

- Registering and unregistering for notification
- How to use a few of the ASFile methods

### Caveats

This code doesn't attempt to associate a print modifying file with a particular open viewer file. The method for disabling the added PostScript isn't too obvious. A better approach might involve use of a popup menu which allows you to specify which file, and an enable/disable item.

Some applications will attempt to clear a page before printing by painting the entire page white at the beginning of the page description. In this case Postscript added at the beginning of the page will not be displayed as will be painted white.

### Source

- PLUGINS:MAC:SAMPLES:AddPS
- PLUGINS\WIN\SAMPLES\ADDPS
- PLUGINS/UNIX/SAMPLES/ADDPS

## 21 Balloon

This plug-in creates a new action type for links. When a link is created a new action can be defined called “balloon help”. When a link with a balloon help action is clicked it will invoke the balloon help plug-in and display an alert with a message which has been stored in the link. The user can edit this message when they create the link by choosing “edit” when they define the action, or by selection properties for the link. By editing the action they will be prompted for text to be displayed in the alert.

Balloon also demonstrates importing an HFT from another plug-ins. Comments describe several approaches to this.

### Illustrates

- Defining a new PDAction type; storing and retrieving private COS data

### Source

- PLUGINS:MAC:SAMPLES:Balloon
- PLUGINS\WIN\SAMPLES\BALLOON
- PLUGINS/UNIX/SAMPLES/BALLOON

## 22 DebuggerWindow

This plug-in shows how a plug-in can export its own HFT.

Functionally, it provides a window which can be written to for the purpose of capturing debug statement output. If you are debugging multiple plug-ins, you shouldn’t have to provide debug window support in each single example; let another plug-in handle the job.

### Source

- PLUGINS:MAC:SAMPLES:DebuggerWindow
- PLUGINS\WIN\SAMPLES\DEBUGWIN
- PLUGINS/UNIX/SAMPLES/DEBUGWIN

## 23 HFTQuery

Demonstrates how to communicate with the Search plug-in through the Search plug-in's HFT. Shows how to set the word options and number of documents, to add and remove indices, and to initiate a search query.

### Source

- PLUGINS:MAC:SAMPLES:HFTQuery
- PLUGINS\WIN\SAMPLES\HFTQUERY

## 24 ImageSel

Implements a selection server, so that images can be selected on the page. This plug-in installs a button on the toolbar. When used, the cursor changes to an image selection icon. The user can click on an image on the screen. Currently, the image will then be surrounded by a gray box. Later implementations might allow you to copy the image to the clipboard, or offer other manipulations of the image.

### Source

- PLUGINS:MAC:SAMPLES:ImageSel
- PLUGINS\WIN\SAMPLES\IMAGESEL
- PLUGINS/UNIX/SAMPLES/IMAGESEL

## 25 ModPDF

This plug-in demonstrates how to modify the page contents of a PDF file. The example draws a black square in the lower left corner of the first page of the open document. It adds a new font to the PDF file, then draws text in the new font and a rectangle around the text. ModPDF also manipulates the current transformation matrix (CTM) to achieve page rotation-independent positioning of page markings.

### Illustrates

- Enclosing modification to page contents in save/restore blocks.
- Forcing a screen update after modifying page contents.

### Source

- PLUGINS:MAC:SAMPLES:ModPDF
- PLUGINS\WIN\SAMPLES\MODPDF
- PLUGINS/UNIX/SAMPLES/MODPDF

## 26 Notify

This plug-in keeps track of the event notifications possible in the viewer. It registers a procedure for each notification. When called, each procedure increments a counter. When the Event Statistics menu item is selected from the File menu, the current counts for each event type are displayed.

### Illustrates

- Event notification registration

### Source

- PLUGINS:MAC:SAMPLES:Notify
- PLUGINS\WIN\SAMPLES\NOTIFY
- PLUGINS/UNIX/SAMPLES/NOTIFY

## 27 OPENALL

This plug-in provides a method for the user to convert non-PDF files on the fly into PDF files which the viewer can display. This plug-in requires PDFWriter and the application which created the file to be present. If a non-PDF file is selected, the creator application is launched and the file is ‘printed’, using PDFWriter. The file created by PDFWriter is displayed by the viewer.

### Illustrates

- Launching another application
- Suppressing the PDFWriter Save As dialog
- Creating a new PDActionHandler
- Replacing a viewer function

### Caveats

This plug-in will not work with all applications. In particular applications which do background printing are not supported.

### Source

- PLUGINS\WIN\SAMPLES\OPENALL

## 28 PIStarter

Skeleton C file for a plug-in.

### Source

- PLUGINS:MAC:SAMPLES:PIStarter
- PLUGINS\WIN\SAMPLES\PISTART
- PLUGINS/UNIX/SAMPLES/PISTART

## 29 Progbar

This plug-in demonstrates creation and use of a custom progress monitor.To demonstrate the calls, it walks through each annotation on each page of the file, looking for Link annots.

### Illustrates

- Creating a custom progress bar

### Source

- PLUGINS:MAC:SAMPLES:Progbar
- PLUGINS\WIN\SAMPLES\PROGBAR

## 30 RFS

This plug-in shows how to create a simple replacement file system.

### Source

- PLUGINS:MAC:SAMPLES:RFS
- PLUGINS\WIN\SAMPLES\RFS
- PLUGINS/UNIX/SAMPLES/RFS

## 31 RplcDemo

Rplcdemo replaces the AVAlertNote method, appending the string “rplcdemo was here” to the passed string parameter, then sending this new string out via the original AVAlertNote method. It also displays an alert when the viewer first launches in response to the application did launch notification.

### Illustrates

- Replacing a function
- Registering a notification

### Source

- PLUGINS:MAC:SAMPLES:RplcDemo
- PLUGINS\WIN\SAMPLES\RPLCDEMO
- PLUGINS/UNIX/SAMPLES/RPLCDEMO

## 32 SetSec

This plug-in demonstrates setting PDF file security data programmatically using the “Standard” security handler. The Preferences item allows setting passwords and security option defaults.

### Illustrates

- Using the “Standard” security handler
- Setting passwords and security options

### Source

- PLUGINS:MAC:SAMPLES:SetSec

- PLUGINS\WIN\SAMPLES\SETSEC
- PLUGINS/UNIX/SAMPLES/SETSEC

## 33 SnapZoom

This plug-ins creates a new tool, the SnapZoom, which allows you to zoom in on a section of the page. This zoomed view of the page is double the magnification of the page. Releasing the mouse button causes the display to “snap” back to the original view.

Pressing the Option or Control key when pressing the mouse button to activate SnapZoom zooms out, providing 1/2 the magnification.

### Illustrates

- Many AVPageView methods
- Creation and use of a new tool

### Source

- PLUGINS:MAC:SAMPLES:SnapZoom
- PLUGINS\WIN\SAMPLES\SNAP
- PLUGINS/UNIX/SAMPLES/SNAP

## 34 Stamper

These samples implement a rubber stamp tool. This sample illustrates a custom tool and a custom annotation. It also plays a neat sound when you stamp the page. Sound provided only on the Macintosh.

### Illustrates

- Custom tool
- Custom annotation

### Source

- PLUGINS:MAC:SAMPLES:Stamper
- PLUGINS\WIN\SAMPLES\STAMPER
- PLUGINS/UNIX/SAMPLES/STAMPER

## 35 SubCrypt

This plug-in demonstrates how to use the security API to implement an application that depends on encryption but not passwords.

This is really two plug-ins in one: a publisher plug-in that encrypts documents and computes subscription information and a subscriber plug-in that opens encrypted documents. A document is encrypted using a key that is based on the ‘magazine’ name, volume, and issue. The name, volume,

and issue are stored as clear text in the file. However, a user cannot change this information because the key computed to decrypt the rest of the file will become invalid.

A subscriber can open a file if she has a valid subscription.This plug-in keeps a file containing a series of pairs of lines separated by carriage return linefeed pairs. The first line of a pair is the name of a magazine. The second is an encrypted range of valid volumes and issues.

The plug-in decrypts the subscription range to determine if the subscription includes the current document. If not, the user is given a chance to enter a new subscription. In real life, the user would have to call a publisher to get a new subscription code, or in a future world, the plug-in could connect by modem to directly obtain the code.

The plug-in provides a command that computes a subscription code. It also allows a user to switch between publisher and subscriber mode. In publisher mode, a document can be opened without a valid subscription.

### Illustrates

- Implementing a custom security handler

### Source

- PLUGINS:MAC:SAMPLES:SubCrypt
- PLUGINS\WIN\SAMPLES\SUBCRYPT

## 36 Template

Template is a plug-in “starter” which follows plug-in UI style guidelines

Template has:

- a menuitem which brings up an alert
- a toolbutton which brings up the same alert as the menuitem.
- The toolbutton is only installed if there is room on the toolbar.
- The command menuitem and toolbutton are enabled always, but a #define can change them to be enabled only when there is an open document.
- About, Help, and Edit/Preferences menuitems, each properly identifying the plug-in according to the style guide, without extra text (e.g.,”About”)
- Export, Import, Init, and Unload callbacks

### Caveats

Toolbutton isn't associated with a tool.

### Source

- PLUGINS:MAC:SAMPLES:Template
- PLUGINS\WIN\SAMPLES\TEMPLATE
- PLUGINS/UNIX/SAMPLES/TEMPLATE

# 37 TextFloaters

This plug-in allows the user to put a floating window over the document. It installs a Create Floater item to the plug-ins menu. Each time this item is selected, a floating window is displayed. The text in the window can be edited, and the window moved around, resized, etc. This floating window remains on the screen independent of the document under it. The document can be moved around, scrolled, resized, etc. On the Macintosh, the window accepts text and standard editing features like cut, copy, paste, etc.

### Illustrates

- Interaction with the underlying windowing model
- AV method usage
- Creation and use of an AVWindowHandler
- Conversion of coordinate spaces between AV space and platform graphics space
- Only the Macintosh uses AVWindow methods

### Source

- PLUGINS:MAC:SAMPLES:TextFloaters

# 38 Wordfind

This plug-in exercises the PDWordFinder methods. The user can build a page map which contains the offset from start of file for each word in the file. Once this is created, the user can tell the plug-in to display and highlight a word in the file, given its offset.

This plug-in can also be compiled to build page maps based on character offsets instead of word offsets. In either case the example will work the same, but the character offset option is provide for systems which prefer to store character offset information instead of word offsets.

Two items are added to the Extensions menu: Create Page Map, and Find Word By Offset. The page map file will have an extension of ".map" and the same root filename as the open document. The Find Word By Offset item displays a dialog in which the user types a number. This number is looked up in the page map and the corresponding word scrolled to and the word is highlighted.

### Illustrates

- PDWordFinder

### Source

- PLUGINS:MAC:SAMPLES:WordFind
- PLUGINS\WIN\SAMPLES\WORDFIND
- PLUGINS/UNIX/SAMPLES/WORDFIND