Release Notes for Adobe® Acrobat® 2.1 Plug-Ins SDK

1 About these Release Notes

These Release Notes tell you about the Acrobat Plug-Ins SDK Release 2.1:

- New features in Acrobat and the Acrobat SDK
- Compilers supported in the SDK
- Problems fixed in this release
- Known problems in this release

2 New Features

2.1 Acrobat Products New Features

New Adobe Acrobat product versions include Acrobat 2.1 for Apple® Macintosh® and Microsoft® Windows®, and Acrobat Distiller 2.1 for Macintosh and Windows.

Some specific features of these new versions:

- Windows Acrobat 2.1 now works under Windows NT™ and Windows 95. Acrobat is a 16-bit application, however.
- Acrobat Exchange LE is obsolete: Exchange LE 2.0 was the last version. Users of Exchange LE should, in general, switch to Exchange. Some may find Reader sufficient.
- The new Weblink plug-in adds the capability of linking to documents on the World Wide Web through your Web browser. Weblink's API allows other applications and plug-ins to call Weblink.

- There are new notifications, methods, and error codes for Macintosh and Windows. See *Acrobat Viewer Plug-In API On-line Reference* for details.
- The Text Extraction ToolKit has been renamed the *Acrobat Toolkit*. This ToolKit is no longer free; contact the Adobe Developers Association at 415-961-4111 for information on licensing it. The Toolkit is available for Windows NT (Win32), IRIX™, HP-UX, SunOS™, Solaris®, and AIX®.
- The SGI Acrobat Reader 1.0.1 is now bundled.
- Several new methods have been added.

  PDModel:
  PDPageNotifyContentsDidChangeEx()
  Broadcasts a PDPageContentsDidChangeEx notification, which notifies the Acrobat viewer that a page's contents have been modified, and tells the Acrobat viewer whether or not to redraw the page immediately.

  PDDocClearFlags()
  Clears flags associated with a document.

  PDDrawCosObjToWindow()
  Draws a specified stream of PDF marking operators into a specified window.

  AVModel:

  AVDocGetClientName()
  Gets (sets) the AVDoc client (OLE container application) name.

  AVDocGetPageText()
  Gets the text from the specified text selection, converts it to a specified format, and passes it to a user-provided procedure.

- Some changes are required to use these new Acrobat 2.1 APIs listed above:
  - *PIRequir.h* must be changed for all platforms. To use a PDModel method, for instance, change the line

    ```
    #define PI_PDMODEL_VERSION ((2L<<16) + 0)
    ```

    to

    ```
    #define PI_PDMODEL_VERSION ((2L<<16) + 1)
    ```

    To use the new AVModel methods, change

    ```
    #define PI_ACROVIEW_VERSION ((2L<<16) + 0)
    ```

    to

    ```
    #define PI_ACROVIEW_VERSION ((2L<<16) + 1)
    ```

These new lines are already provided in the file as comments.

- For the Macintosh, you must also do the following:

  Precompile headers for both 680X0 and PPC (using a different name for the new precompiled header).

  Modify the project to include the updated precompiled header file.

- Once you've made these changes, you can't run the newly built plug-in in the Acrobat 2.0 viewer.

2.2 Acrobat Plug-ins SDK New Features

The Acrobat Plug-ins SDK sports new features in these areas:

2.2.1 General

- Plug-in developers can access a World Wide Web site that contains the latest updates to the Acrobat Plug-ins SDK:
  http://www.adobe.com/acropid
  Contact the Adobe Developers Association for your login and password.
- The SDK has been reorganized for easier access. *HOMECARD.PDF* has been replaced by *WELCOME.PDF* and provides a more logical grouping of files, as does the new file structure on the CD.
- The Plug-ins API and Interapplication Communication Technical Notes have each been split into conceptual descriptions and on-line guides.
- Acrobat 2.0 products are included in the *PRODUCTS:MAC:OLD* and *PRODUCTS\WIN\OLD20* directories. This software is provided *only* for backward compatibility testing.
- Symantec® Think C™ 7.0.4 is no longer supported.

2.2.2 IAC

- The OLE samples that use the Draw() method, Draw and VBDraw, for Visual C++ and Visual Basic respectively, have been renamed DRAW and VBDRAW to reflect their use of Draw(). Similarly, the samples that use the OpenInWindow() method, OIW and VBOIW, have been renamed OIW and VBOIW to indicate that they use OpenInWindow().
- OIW: Windows Distiller manipulation is now part of the OIW application. The Distiller sample is now separate from OIW.
- VBOIW includes more features.
- CAPTURE is a new 32-bit sample that shows communication with Capture using DDE.

- PDFWCTRL is a new Windows application that demonstrates how to communicate with PDFWriter.
- DDEOPEN is a new, simple DDE example that demonstrates how to open a file through DDE. If you are going to use DDE to communicate with the Acrobat viewer, you can start with this sample and build up your application from there.
- The AEView sample has been improved to communicate with Acrobat Search through Apple events to add indices and initiate search queries. It also demonstrates how to communicate with PDFWriter and Distiller.

2.2.3 Plug-ins

- UNIX header files, built plug-ins, and samples are now included for SunOS, Solaris, and HP-UX systems. There are no UNIX samples for HFTQUERY, PROGBAR, SUBCRYPT, or TEXTFLTR.
- RFS: This new plug-in sample demonstrates basic file system replacement.
- Precompiled headers are not included with the SDK. See *Acrobat Viewer Plug-In API Development* for precompiling instructions.
- The TextFloaters plug-in sample has been moved to the *UNSUPPTD* directory for Windows since only the Macintosh version uses AVWindow methods.
- The Technical Note # 5167, *Acrobat Viewer Plug-In API Development*, tells how to develop plug-ins on Power Macintosh® and 680X0 Macintosh, Windows, and UNIX® platforms. It presents information on registering and using plug-in prefixes. This Note also gives general advice on plug-in development and porting among platforms.
- The Acro21 plug-in describes how to build an Acrobat 2.1 plug-in and how to use the 2.1 HFT calls. If you use the Acrobat 2.1 HFT methods, your plug-in will not work under Acrobat 2.0 or 2.0.1.
- Both Power Macintosh and 680X0 plug-ins are supported. All SDK code samples demonstrate how to create a “fat” plug-in that includes both a 680X0 code resource and a PowerPC™ code fragment and can run on either a 680X0 Acrobat 2.0 or 2.1 viewer or a Power Macintosh Acrobat 2.1 viewer.
- HFTQuery for Macintosh and Windows: Demonstrates how to communicate with the Acrobat Search plug-in through the Search plug-in’s exported HFT.
- The ImageSel plug-in for Macintosh and Windows now sends its selection information to the debugger window, allowing developers to see the order in which the selections occur. The DebuggerWindow plug-in must be installed for this to work.

- Notify for Macintosh and Windows now sends its notifications to debugger window, allowing developers to see the order in which the notifications occur. The DebuggerWindow plug-in must be installed for this to work.
- The SetSec plug-in manipulates the security settings for a document using the API. It allows batch setting of security on PDF files.
- The Stamper plug-in sample for Macintosh and Windows now sends its selection information to the debugger window, allowing developers to see the order in which the annotation's methods occur. The DebuggerWindow plug-in must be installed for this to work.
- The Compound (Comp) file system replacement plug-in for Macintosh and Windows has been moved to the unsupported sample directories.

3 Supported Compilers

The Acrobat SDK supports the following compilers:

- Microsoft Windows™ Visual C++ 1.5.2
- Microsoft Windows Visual Basic 4.0
- MetroWerks CodeWarrior 6.0 and 7.0
- Macintosh AppleScript™ 1.1

Microsoft Visual C++ and MetroWerks CodeWarrior project files are included with the SDK samples.

4 Problems Fixed in Version 2.1

The following problems have been fixed in Acrobat 2.1.

4.1 Acrobat Viewers

4.1.1 Macintosh and Windows

Description: The HFT replaceable plug-in method AVDocClose() displays a Save As dialog box whether or not it is replaced. This error still occurs, but you can now clear flags that have been set to indicate that the file is dirty. Use the Acrobat 2.1 function PDDocClearFlags() to clear these flags.

4.1.2 Macintosh

1. **Description:** The Apple event DrawPageToWindow now draws correctly to an off screen GWorld window pointer. Use the optional boolean parameter keyAEThisIsAGWorld in the DrawPageToWindow event.
2. **Description:** The Apple event for creating thumbnails now works correctly.
3. **Description:** The API method PDPageDrawContentsToWindow() now draws text correctly on pages that have been rotated in the viewer. Previously, it drew the text greeked on pages that had been rotated in the viewer unless the user set the greeking value to –fixedOne.

4.1.3 Windows

1. **Description:** The OpenInWindow() OLE call now supports the various menu items that are available when a file is open in the viewer. You can both check their availability and execute them. Exchange 2.0 was not able to execute certain menu items because OpenInWindow() did not activate menu items that should be active when a file is open. Therefore, menu items such as Copy, Paste, and Select All, which are active when a file is open, were not available in Exchange 2.0.
2. **Description:** Developers using the Acrobat OLE command Close() used to encounter a GPF if the document had been dirtied, a noSave of FALSE was passed to the Close() method, and the user had to click the Cancel button rather than Yes or No. This now works correctly.
3. **Description:** A problem relating to deleting an OLE PDPage object has been fixed. The OLE calls that now work but previously failed are these:

- Create a new AcroExch.App object
- AcroExch.App.Lock()
- Create new AcroExch.AVDoc object
- AcroExch.AVDoc.OpenInWindow()
- AcroExch.AVDoc.GetAVPageView()
- AcroExch.AVPageView.GetPage()

- Free a AcroExch.PDPage object to delete it (previously resulted in a GPF)

4.2 Acrobat Search

4.2.1 Windows

Description: Using a language type of Simple Query to communicate with Acrobat Search no longer results in a GPF.

5 Known Problems

This section lists all known problems. If you find a problem that is not listed, see the file *SUPPORT.PDF* in the SDK for information on obtaining support.

Note *Also see the Caveats sections in the SAMPLES.PDF file for important information about possible problems.*

5.1 Windows NT and Windows 95

Description: The Acrobat Exchange 2.1 executable for Windows is a 16-bit application that works under Windows NT and Windows 95. (Exchange 2.0 did not work with Windows NT.) If you are developing a 32-bit application to communicate with the Acrobat viewer, there are some 32- to 16-bit message conversion problems. In particular, the Acrobat OLE command Draw fails because the hDC passed to it from the 32-bit application is mangled incorrectly. The Acrobat OLE command OpenInWindow() works correctly; however, take note of the modified OIW example. You must add message handling for OnEraseBkgnd() and OnSize(). Here is the code you must implement for both of these functions:

```
BOOL CDrawView::OnEraseBkgnd(CDC* pDC)
{
   /* lets pretend we already erased the background */
   return 0;
}

void CDrawView::OnSize(UINT nType, int cx, int cy)
{
#ifdef WIN32
   CWnd* pWndChild = GetWindow(GW_CHILD);
   if (!pWndChild)
      return;
```

```
    CRect rect;
    GetClientRect(&rect);
    pWndChild->SetWindowPos(NULL,0,0,rect.Width(),rect.Height(),
      SWP_NOZORDER|SWP_NOMOVE);
#endif
    CView::OnSize(nType, cx, cy);
}
```

If you are writing a 16-bit application for Windows NT or Windows 95, you do not have to be concerned about issues regarding messages passed between your application and the Acrobat viewer. A 16-bit application can communicate with the 16-bit viewer with no mangling problems.

5.2 Acrobat Viewer IAC

5.2.1 Windows

1. **Description:** PrintPages() generates a GPF 0007:2CFF if you use a modal dialog.

   Workaround: Make the print dialog box modeless.

2. **Description:** The Visual Basic sample application VBDraw does not display files properly with Visual Basic 3.0 but does work with Visual Basic 4.0.

   Workaround: Use Visual Basic 4.0.

3. **Description:** The Visual Basic sample application VBOIW gets a GPF when trying to print files.

   Workaround: Make the print dialog box modeless.

5.3 Acrobat Viewer API

5.3.1 Macintosh and Windows

1. **Description:** You cannot use either the plug-in API calls PDWordGetAttr() or PDWordGetCharacterTypes() to determine if a particular PDWord or character is uppercase.

   Workaround: Check directly if characters are uppercase with code such as this:

```
boolean hasUpper(PDWord pdWord)
{
   char buf[128];
   Uns8 i, tmp;
   PDWordGetString(pdWord, buf, sizeof(buf));
   tmp = PDWordGetLength(pdWord);
   for(i=0;i<tmp;i++){
```

```
      if(buf[i] >= 'A' && buf[i] <= 'Z') break;
    }
    if (i == tmp)
      return false;
    else
      return true;
  }
```

2. **Description:** Using the plug-in API call AVDocPrintPages() or an IAC method that calls this method results in a dialog box that says "Printing Pages *x* of *y*."

   Workaround: None. There is no way to avoid displaying this dialog box.

3. **Description:** The plug-in API calls PDDocCreateThumbs() and PDDocDeleteThumbs() ignore the start and end page numbers.

   Workaround: None. You can only create or delete *all* thumbnails.

4. **Description:** The plug-in API call PDGraphicEnumMonitor() continues enumerating whether the enumerator returns false or not.

   Workaround: None. You must wait for the graphic enumerator to finish.

5. **Description:** If you replace the AVDocDoSave() method in the HFT, you are not called if the user clicks the "Save As" button.

   Workaround: Register for the PDDocWillSave notification.

5.3.2 Macintosh

1. **Description:** The following are notification and replaceable method problems:

   - The FrontDocDidChange and DocDidActivate notifications occur before the DocDidOpen notification.
   - The DocDidOpen notification occurs even if you are opening a document that is already open.
   - You do not receive a DocDidOpen notification if you do a Command-? to open the Acrobat Exchange Help file.
   - If you replace CanQuitProcedure() in the HFT, all of the documents are closed before your replaced function is called.

5.3.3 Windows

1. **Description:** All the OLE 2.0 files were not installed with the Acrobat viewer. The files missing are bundled with the SDK in the \SOURCE\IAC\WIN\SDK directory.

   Workaround: See the *README* file in that location for details on correctly installing and registering all OLE files and methods.

2. **Description:** Executing the menuitem “Find text” and searching causes a GPF if the text is not found.

   Workaround: None.

3. **Description:** Developers creating their own custom file system cannot use the Acrobat OLE method OpenInWindow(). OpenInWindow() automatically uses the default file system.

   Workaround: Use the PD level OLE method Draw() instead. However, Draw() does not give you all the capabilities of OpenInWindow().

4. **Description:** Cross document links do not work when a file is opened with the Acrobat OLE command OpenInWindow().

   Workaround: Use internal links only within PDF documents when using the OpenInWindow() method.

5. **Description:** When using either ASMemStmRdOpen() or ASFileStmRdOpen(), you are restricted to creating streams of 64K (65535 bytes) or less.

   Workaround: Use the method ASProcStmRdOpen() to create a stream procedure that defines a HugePtr to store the data. Here is an example of such a procedure and a structure to store the data:

```
typedef struct {
   char __huge *ptr;
   Int32 pos;
   Int32 sizedata;
   } MyProcData;

static ACCBPROTO1 Int32 (ACCBPROTO2 *myASStmProc)
   (char *data, Int32 nData, void *clientData);

static ACCB1 Int32 ACCB2 myASStmFunc(char *data,
   Int32 nData, void *clientData)
{
   MyProcData *myData = (MyProcData *) clientData;
   Int32 nBytes;

   nBytes = min(nData, myData->sizedata);
   memcpy(data, myData->ptr + myData->pos, nBytes);
   myData->pos += nBytes;
   myData->sizedata -= nBytes;

   return nBytes;
}
```

6. **Description:** Acrobat 2.1 for Windows does not support the OLE 2.0 method OleCreateFromTemplate().

   Workaround: In the Lotus Notes example code, there is a demonstration using OleLoad() to load an OLE object instead of OleCreateFromTemplate().

7. **Description:** Shortcut keys for menu items cannot be reused once a menu item has been added. For instance, suppose you remove the Print menu item with the Control-P shortcut key and add a new menu item that uses Control-P. When Control-P is pressed, the old menuitem's execute procedure is called—not the new one's.

   Workaround: None.

8. **Description:** If you replace AVDocDoSave() in the HFT, the replaced method is not called if the user closes a dirtied document or quits the viewer with a dirty document that is open.

   Workaround: Replace the AVDocClose() and AVAppCanQuit() methods in the HFT and place the clean-up code in the methods you replace them with.

5.4 Acrobat Search

5.4.1 Macintosh and Windows

1. **Description:** If you are communicating with the Search plug-in through DDE or Apple events, you must specify a MaxDocs value that encompasses the total number of documents that can be found. You cannot restrict the number of documents found with the MaxDocs parameter.

   Workaround: None.

2. **Description:** There is an inconsistency between the Macintosh and Windows Search plug-ins. The Windows version lets you specify a sortways argument that lets you specify an ascending or descending sort. The Macintosh always defaults to descending (highest hit value at top).

   Workaround: None.

5.5 Distiller™

5.5.1 All platforms

Description: If you have a document containing pdfmark information that supports Acrobat 2.0 features, and you are using Distiller 1.0, it reports a PostScript® error.

Workaround: None.

5.5.2 Windows

Description: With Acrobat Distiller 2.0, using the /o option to specify a destination file results in a PDF file with an extension of *.PDF* regardless of which extension was specified. Using the /o option to specify a list of PostScript language files does not work with the Acrobat 2.0 Distiller.

Workaround: None.

5.6 Acrobat PDFWriter

5.6.1 Windows

Description: The PDFINPUT structure members PDFINPUT.username and PDFINPUT.creator are ignored by PDFWriter 2.0 when using the PDFFILENAME printer escape.

Workaround: None.

5.6.2 Macintosh

1. **Description:** If you specify the root directory as the output path, you may crash.

   Workaround: Don’t write to the root directory. If you have written to the root directory and crashed, reinstall PDFWriter.

   Description: The AEView sample has an output path that may not exist.

   Workaround: Change the path appropriately for the output file in the sample.

5.7 Acrobat Catalog

5.7.1 Windows

Description: Catalog does not respond to the Windows message AcrobatCatalogQueryHello.

Workaround: None.