



PDF Creation Date:

September 1, 2006

Adobe PDF Library Overview

Copyright 2006 Adobe Systems Incorporated. All rights reserved. Adobe® PDF Library Overview.

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

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

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

PostScript is a registered trademark of Adobe Systems Incorporated. All instances of the name PostScript in the text are references to the PostScript language as defined by Adobe Systems Incorporated unless otherwise stated. The name PostScript also is used as a product trademark for Adobe Systems' implementation of the PostScript language interpreter.

Except as otherwise stated, any reference to a "PostScript printing device," "PostScript display device," or similar item refers to a printing device, display device or item (respectively) that contains PostScript technology created or licensed by Adobe Systems Incorporated and not to devices or items that purport to be merely compatible with the PostScript language.

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

Apple, Macintosh, and Power Macintosh are trademarks of Apple Computer, Inc., registered in the United States and other countries. PowerPC is a registered trademark of IBM Corporation in the United States. ActiveX, Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. UNIX is a registered trademark of The Open Group. All other trademarks are the property of their respective owners.

This publication and the information herein is furnished AS IS, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied, or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes, and noninfringement of third party rights.

Portions include software under the following terms:

Portions derived from the RSA DataSecurity, Inc. MD5 Message-Digest Algorithm.

This product contains either BISAFE and/or TIPEM software by RSA Data Security, Inc.

Portions of the software were developed by the University of California, Berkeley.

The author of this software is David M. Gay. Copyright (c) 1991 by AT&T. Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR AT&T MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

Portions of the software were developed at Cygnus Solutions.

Portions copyright (c) 1995-2003 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

Portions copyright (c) 1994 Hewlett-Packard Company. Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Hewlett-Packard Company makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright (c) 1996 Silicon Graphics Computer Systems, Inc. Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Silicon Graphics makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

The TWAIN Toolkit is distributed as is. The developer and distributors of the TWAIN Toolkit expressly disclaim all implied, express or statutory warranties including, without limitation, the implied warranties of merchantability, noninfringement of third party rights and fitness for a particular purpose. Neither the developers nor the distributors will be liable for damages, whether direct, indirect, special, incidental, or consequential, as a result of the reproduction, modification, distribution or other use of the TWAIN Toolkit.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA. Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contents

Preface	4
What's in this guide?	4
Who should read this guide?	4
Related documentation	4
1 Introduction	5
The PDF Library and the PDF viewers	5
New in version 8	6
2 About the PDF Library SDK	7
Supported environments	7
Contents	7
SnippetRunner	7
Installing the PDF Library SDK	13
Updating to PDF Library Version 8	13

Preface

This document provides an overview of the Adobe PDF Library SDK.

What's in this guide?

This guide provides an introduction to development using the Adobe PDF Library. It describes the Adobe PDF Library, the Adobe PDF Library Software Development Kit (SDK), support and licensing options, and basic development tasks.

Who should read this guide?

This guide is intended for programmers who need basic information on the Adobe PDF Library and its capabilities. It assumes familiarity with C programming, common development tasks, and the use of methods exported by an object code library.

Related documentation

The following resources provide further information about the PDF Library SDK, as well as additional documents that you should have available for reference.

For information about	See
A detailed description of the PDF file format.	<i>PDF Reference</i>
Developing plug-ins for Acrobat and Adobe Reader, as well as for PDF Library applications.	<i>Developing Plug-ins and Applications</i>
Detailed descriptions of the APIs for Acrobat and Adobe Reader plug-ins, as well as for PDF Library applications.	<i>Acrobat and PDF Library API Reference</i>
An overview of the SnippetRunner tool and the plug-in snippets provided with the Acrobat SDK.	<i>SnippetRunner Cookbook</i>
Licensing and other information about the PDF Library SDK.	www.adobe.com/go/acrobat_developer
Information about Adobe enterprise developer resources.	www.adobe.com/enterprise/developer

Designed specifically for OEMs, ISVs, system integrators, and enterprise IT developers, the Adobe PDF Library SDK contains a powerful set of functions for developing third-party solutions and processes around Adobe PDF. The PDF Library is based on the core technology of the Adobe Acrobat line of products and offers complete functionality for generating, manipulating, rendering, and printing PDF documents.

The library enables PDF functionality to be seamlessly embedded within applications. It also provides reliable, accurate and Adobe-supported implementation of the latest PDF specification.

The PDF Library and the PDF viewers

The PDF Library shares much of the same source code as the Adobe PDF viewing applications (Acrobat Professional, Acrobat Standard and Adobe Reader). There is a large degree of overlap between the functionality provided by the PDF Library Software Development Kit (SDK) and that of the Acrobat SDK. They differ in providing access to the Acrobat user interface:

- The Acrobat SDK is meant for the plug-in environment, and allows you to control and interact with the Acrobat user interface.
- The PDF Library SDK is intended for interaction between PDF content and other applications, such as high volume batch processing and PDF generation applications. It does not export methods for creating or managing Acrobat UI elements—that is, the AcroView (AV) layer of the core API.

If you are interested in the documentation and samples in the Acrobat Software Development Kit (SDK), see http://www.adobe.com/go/acrobat_developer.

New in version 8

The version 8 release of the PDF Library offers support for PDF 1.7 features. These features are described in detail in the *PDF Reference*; Chapter 1 of that document contains a list of the new features.

The PDF Library includes API methods for creating PDF files that take advantage of new PDF 1.7 features. It supports the same feature set as the Acrobat 8 API, except for those features directly related to the user interface (that is, the AV layer and plug-in support).

The version 8 release also provides new snippets. The section [“SnippetRunner” on page 7](#) lists the snippets and indicates which ones are new in version 8.

This chapter helps you get started with development using the PDF Library SDK. It describes the contents of each directory in the SDK installation, lists available code samples, and provides platform-specific information on how to set up the development environment.

Supported environments

PDF Library 8.0 and its project files are supported for the platforms, operating systems, and compilers listed in *Developing Plug-ins and Applications*. While it may be possible to use the library in other development environments, such use is not supported by Adobe Developer Support.

Contents

The PDF Library 8.0 SDK consists of the following components:

- Core libraries that provide the functionality
- Header files that provide access to the libraries
- Fonts used in the library's basic operations
- Sample applications and code snippets showing how to use the library for a variety of purposes
- Documentation of installation and development techniques and of the PDF Library APIs

The components that are shipped with the PDF Library 8.0 SDK are listed in *Developing Plug-ins and Applications*. These include the PDF Library, helper libraries, headers, and samples.

SnippetRunner

The SnippetRunner is based on the SnippetRunner provided with the Acrobat 8 SDK, but is adapted to the PDF Library environment. It has a command-line interface and prints output to the console. It is found in the SnippetRunner subdirectory of the Samples directory. For more information, see the *SnippetRunner Cookbook*.

The following table shows the commands that you can use to control the snippet runner in a command shell. The snippets (see [“PDF Library snippets” on page 8](#)) are arranged in a directory-like hierarchy. You can use the `findsnip` command to locate a snippet and the `cd` command to switch between directories.

Note: The command names are not case-sensitive. However, the snippet names and directory names are case-sensitive.

Snippet Runner commands

Command	Description
<code>cd</code>	Navigate the hierarchy of snippets using standard file path semantics.
<code>findsnip</code>	Search for a snippet based on a specified substring of the snippet identification string. If the result is a single file, prompts for execution.

Command	Description
snipinfo	Display the description of a specified snippet.
docname	Display the name and file information for the current document.
setpage	Specify the current page of the document.
opendoc	Open the specified document. This document becomes the new current document. If the previously open document was modified, the command prompts to save changes.
closedoc	Close the current document. If it was modified, the command prompts the user to save changes.
savedoc	Save the document to the specified file.
revert doc	Revert the current document to its last saved condition.
run	Run the specified snippet. The snippet must exist in the current directory—you cannot specify a path.
ls	List the snippets (and hierarchy nodes) in the current directory.
pwd	Print the current directory.
help	List all the commands with their synopses, or provide help for a specified command.
quit	Quit the application.
whichPage	Display the current selected page.

The snippets provided include most of those available for the Acrobat SDK. Snippets include the following:

PDF Library snippets

Snippet	Description
ACEEnumProfilesSnip	Enumerates the profiles of a given type installed on the system. Makes a profile list, gets a count of how many profiles are in the list, gets the description for each item in the list, and dumps the information in the DebugWindow.
ACEEnumSettingsSnip	Demonstrates how to programmatically access color setup files using the Adobe Color Engine (ACE).
ACEGetWorkingSpaceSnip	Gets working space profile (RGB, grayscale, and CMYK) information such as size, description and color spaces , and lists the information in the DebugWindow.
ACETransPDETextColorSnip	Takes an input profile (DeviceCMYK) and an output profile (sRGB), creates a color transform from them, and then applies the transform to a single color of a PDF text object.
AddImageMetadataSnip	Shows how to add XMP metadata to an image XObject in a PDF document.

Snippet	Description
AddImageSnip	Creates an image XObject resource from a JPEG file and adds it to the displayed page.
AddPageMetadataSnip	Shows how to add XMP metadata to a page in a PDF document.
AddStructureSnip	Shows how to work with logical structure within a PDF document. Tags PDEText elements within the displayed page.
AddTagSnip	Shows how to add a partial structure tree to an un-tagged PDF document, add tags, and add three types of content items (marked content, structure elements, and complete PDF objects).
AddTextSnip	Adds a string of English, Chinese, Korean, or Japanese text to the bottom left corner of the current viewing page. It demonstrates font embedding/subsetting and support for CJK double-byte fonts in Acrobat. Fonts used in this sample are Times-Roman for English, SimSun for Chinese, Batang for Korean, and MS Mincho for Japanese.
AddXObjectStructureSnip	Shows how to work with logical structure within a PDF document. Tags image XObjects within the displayed page.
ASBigFileSnip	Demonstrates how to manipulate large files.
ASCabPutGetSnip	Demonstrates how to put arbitrary key-value pairs into an ASCab object and how to retrieve the corresponding value for a specified key from the ASCab object.
ASChangeTempFileSysSnip	Demonstrates how to change the temporary file system implementation for a platform.
ASDateSnip	Gets today's date, and adds different time spans to a date.
ASFileIteratorSnip	Iteratively visits all files in a given folder and its sub-folders, outputting the file names to the DebugWindow.
ASGetConfigurationSnip	Checks for the current configuration of Acrobat, and displays an alert with the return value.
ClassMapSnip	Demonstrates that an application using the PDF Library that processes logical structure can attach additional information (attributes) to any structure element.

Snippet	Description
ColorSetupSnip	<p>Shows how to programmatically access the color management settings using Adobe Color Engine (ACE). It enumerates all available system color settings, ICC profiles, and conversion engines.</p> <p>For example:</p> <p>On Windows 2000:</p> <ul style="list-style-type: none">C:\Program Files\Common Files\Adobe\Color\ProfilesC:\WINNT\system32\spool\drivers\color <p>On Mac OS:</p> <ul style="list-style-type: none">System Folder:ApplicationSupport:Adobe:Color:Systems Folder:ColorSync Profiles:
ConvertOCGsToRadioButSnip	<p>Creates a radio-button relationship among the OCGs in the document, so that only one OCG in the document can be displayed at any one time.</p>
CosCryptGetVersionSnip	<p>Reports the version of the current encryption algorithm, and the maximum number of bytes that can be used for the key when encrypting or decrypting with this version.</p>
CosDoc64Snip	<p>Demonstrates 64-bit file position access into CosDoc objects.</p>
CosObjCompressionSnip	<p>Shows the use of new Cos layer methods to perform full compression of indirect objects in a document to reduce PDF file size.</p>
CosObjDecompressionSnip	<p>Shows the use of new Cos layer methods to perform decompression of compressed indirect objects in a document so as to restore backward viewer compatibility of the document.</p>
CosObjectExplorerSnip	<p>Prints out either a shallow or a more complete description of a Cos object.</p>
CosStream64Snip	<p>Demonstrates 64-bit APIs for Cos stream object construction and info access. The snippet creates a new PDF document with an embedded image taken from an input file represented as a Cos stream.</p>
CreateAnnotOCSnip	<p>Create an optional content group and associates it to any link annotations found on the first page of the front document through an optional content membership dictionary.</p>
CreateContentXORSnip	<p>Shows an exclusive OR relationship between optional content groups. When the content of a page is turned off through the layers panel UI, an alternative text object is turned on.</p>

Snippet	Description
CreateDocStructSnip	Shows how to create a document structure tree that conforms to tagged PDF conventions.
CreateImageContentOCsSnip	Converts images within the first page of the front document to optional content. It iterates through images found on the first page of the document and associates them with a newly created optional content group (OCG).
CreateTextContentOCsSnip	Shows how to convert text into optional text, Converts text within the first page of the front document to optional content.
ExploreMetadataSnip	Shows how to access the XMP metadata stream embedded in a PDF file and write metadata to an XML file.
ExploreStructSnip	Explores the structure and content of a tagged PDF document and dumps the structure information to the console or to the debug window.
FontInfoSnip	Extracts font information from a PDF document, including the name, subtype, and whether the font is embedded.
GetDocKeywordSnip	Extracts the keywords from a document's Info dictionary.
GetDocMetadataSnip	Shows how to extract PDF document XMP metadata in XML format.
ImageInfoSnip	Shows how to obtain specification information of images in a PDF document, including size, width, height, filters, bits per component, and rendering intent.
JPXColorSpaceExplorerSnip	Displays color space characteristics of JPX image XObjects.
JPXPaletteExplorerSnip	Displays palette information of JPX image XObjects.
MakeBookmarkSnip	Makes a bookmark named 'Current Page' for the current page at the top (visible) level of the bookmark tree.
ObjShiftSnip	Demonstrates translating page object positions with PDFEdit APIs.
OCActionControlSnip	Shows how to control the visibility of optional content using PDActions.
OCGUIReorderSnip	Reorders and categorizes the OCGs shown in the layers panel UI.

Snippet	Description
<code>OCTextAutoStateSnip</code>	Shows how to use an autostate with optional content. Creates an optional content group for the text on the first page of the front document. Sets the OCG's usage dictionary to indicate the text should be visible when the zoom factor is between 1.0 (100%) and 1.5 (150%). Updates the document's optional content properties catalog to indicate that the OCG's zoom category should be used to determine visibility for View events. The text disappears when the zoom is out of the specified range.
<code>PDCreateMasterOCGSnip</code>	Creates a parent control OCG for use in the UI. Child OCGs cannot be manipulated through the UI while the parent OCG is Off, although the parent's state does not alter the visibility of the child OCGs.
<code>PDDocDidDeletePagesNotSnip</code>	Shows how to register for and receive <code>PDDocDidDeletePages</code> notifications.
<code>PDEContentExplorerSnip</code>	Writes information about the PDE content for page zero to the <code>DebugWindow</code> .
<code>PDEPathDrawCurveSnip</code>	Demonstrates simple spline curve drawing with <code>PDEPath</code> marking APIs.
<code>PDEPathDrawLineSnip</code>	Demonstrates simple line drawing with <code>PDEPath</code> marking APIs.
<code>PDEPathDrawRectSnip</code>	Demonstrates simple rectangle drawing with <code>PDEPath</code> marking APIs.
<code>PDEPathExplorerSnip</code>	Explores <code>PDEPath</code> objects on the current page by breaking them down to <code>PDEPath</code> marking operators and operand(s).
<code>PDOConfigCreateSnip</code>	Creates an optional content configuration, with the OCGs currently in the ON state assigned the default value of ON, and presenting only these OCGs in the layers panel UI.
<code>PDOConfigExplorerSnip</code>	Uses an optional content group callback to display diagnostic information about the OCGs in a document to the <code>DebugWindow</code> .
<code>PDOCGChangeLockedStateSnip</code>	Toggles the locked state of each optional content group within the document.
<code>PDOCGExplorerSnip</code>	Displays information about optional content configurations to the <code>DebugWindow</code> . Shows the configuration's name, its default state, and lists of OCGs that are on, off and/or locked by default.
<code>PDOCGToggleIntentSnip</code>	Toggles the intent of an optional content group between "design" and "view."
<code>PDOCGSetDefaultConfigSnip</code>	Updates the document's default optional content configuration to use the specified alternative configuration.

Snippet	Description
<code>PDPageSetTransparencySnip</code>	Changes the transparency of the elements in the content of the first page of the current document.
<code>RaiseExcepSnip</code>	Shows how to register custom exceptions with the application. The exception text associated with the custom exception is passed in as a parameter. We use the error code we get from registering the error string to raise an exception. This exception is caught by the <code>SnippetRunner</code> backstop exception handler.
<code>RemoveEmbeddedFontSnip</code>	Removes embedded Roman fonts from a PDF file. An embedded font is removed only if it is not multi-byte, if encoding is not "Identity," and if the charset is Roman.
<code>RoleMapSnip</code>	Enumerates all existing role maps in the PDF document. Also provides an example of how to create new role maps or change existing role maps in a tagged PDF document.
<code>SecureDocumentSnip</code>	Shows how to apply security settings programmatically.
<code>SetDocBaseURLSnip</code>	Shows how to set and get a PDF document metadata property. Sets the base URL from user input.
<code>SimpleSnip</code>	Shows how to create a snippet.
<code>TextChangeColourSnip</code>	Changes the <code>PDEText</code> object of the first page of the front document to specified RGB values.
<code>TextExtractionSnip</code>	Demonstrates how to perform text extraction with the new <code>WordFinder</code> creation APIs and configuration structure.
<code>UserPropertiesExplorerSnip</code>	Demonstrates the use of <code>PDSEdit</code> APIs to enumerate and obtain <code>UserProperties</code> attributes for PDS elements.

Installing the PDF Library SDK

The Adobe PDF Library SDK installation process differs depending on the target platform.

- Windows and Mac OS — An executable installer is provided for each of these platforms. Run the installer and follow the default installation instructions.
- Solaris, AIX, and Linux — A gzip compressed tar file is provided; developers determine where to decompress and install the SDK.

Updating to PDF Library Version 8

All PDF Library SDK 7.0 APIs are still supported in PDF Library 8.0. Other than a few minor code changes, all you should need to do is recompile your application with the new SDK headers using the recommended development environment.

All platforms

Update your font and CMap resources. The following fonts are included with PDF Library SDK 8.0:

- AdobeMingStd-Light (Traditional Chinese)
- AdobeSongStd-Light (Simplified Chinese)
- AdobeMyungjoStd-Medium (Korean)
- KozMinProVI-Regular and KozGoPro-Medium (Japanese)
- AdobePiStd
- CourierStd, CourierStd-Bold, CourierStd-BoldOblique, CourierStd-Oblique
- Symbol
- Adobe Sans MM, Adobe Serif MM

Redirect the path environmental variable as needed, if you do not install the libraries in the same directory as the executable.

Windows

Change `PDFL70.lib` to `AdobePDFL.lib` in your project settings.

Mac OS

Refer to the sample projects for information on updating. PDF Library 8.0 uses frameworks instead of shared libraries. For convenience, the sample applications are built to include the PDF Library frameworks. The universal libraries, which will work on both PPC and Intel platforms, are in the `Libs:Mac:AdobePDFL.framework` folder. Aliases or proper paths need to be set to ensure that your applications run properly.

Include `PDFLInitCommon.c` in the `Include:Headers` folder to your client project. All C source files must be compiled as C++ files using the following steps:

1. In the project source pane, select the file(s) and click Get Info.
2. In the General panel, change the File Type: popup from `sourcecode.c.c` to `sourcecode.cpp.cpp`.

The `Needs Raise Aware` flag has been turned off on Mac OS.

UNIX

Update your makefile to reflect any new libraries in the `libs` folder.