

Release Notes v. 21.0

This guide describes how to install the Adobe PDF Library SDK, as well as general information regarding the release. In addition to this document, please read the [PDF Library documentation](#) for important information on developing with and using the PDF Library.

Tip

This document is available in HTML and [PDF](#).

Version 21.0, February 29, 2023

Changes for this release

Note

The current release does not provide new fonts and the new glyphs of existing fonts. Delivery is expected in a future release. To render newly added glyphs, add your new font files to Resources/font.

- Support for Windows 32-bit systems was dropped
- Compiler changes as described in System Requirements (below)
- Added support on Mac M1 and M2 via XCode 12.4
- Performance fixes for embedded fonts
- Fixed content dropouts at specific zoom levels
- Improvements in the flattening mechanism to resolve text shifts and overlapping issues
- Added support for the Chinese GB18030 font
- PDFL upgraded to OpenSSL_3.0, version 3.0.10 which fixes several vulnerabilities. For details about 3.0, refer to the [Open SSL release details](#).

Note

In our packaging, the updated OpenSSL library is included across all platforms as it is now dynamically linked at runtime.

Fixed bugs

- 4133777: Need to support Apple M1
- 4176667: Unnecessary operands for 'sc' or 'SC' operator by `PDETextRunSetGState()`
- 4176663: ICC Based profiles Alternate color Space is inserted
- 4176645: Text shifts when `PDEFormCreateFromCosObj()` is used

4176642: Printing slows down

- 4176638: Performance degrades while embedding font with 18.0.5
- 4176616: Text shifting after flattening
- 4176607: With **PDFPageDrawContentsToMemory**, some characters disappear on a certain zoom levels.
- 4176606: Text and elements disappear when a PDF is flattened
- 4176605: Arrow color changes when printed
- 4176604: Locating 64-bit and 32-bit versions of the PDF Library in the same folder causes an exception.
- 4176599: Text is shifted
- 4176586: Data lost after splitting the PDF
- 4176531: Marked content without EMC disappears after editing
- 4176482: Text gets scrambled after flattening

Getting started

System requirements

The following operating systems and compilers are supported by PDFL 21.0:

Note

The product dropped support for Windows 32-bit with this release.

V. 21 Compiler Support

Platform	Processor	Compiler	Additional OS
Windows 10 64-bit	Intel	Visual Studio 2019 (16.11.30)	
Linux 64-bit, Ubuntu 18.0.4	Intel	Clang 9	
Mac 64-bit (OS 11.2)	Intel	Xcode 12.4	M1 & M2

Installation overview

All installations generally involve the following:

1. Setting up your environment as described below.
2. Reading the [PDF Overview](#) which provides links to related documentation, including the PDF Reference and Snippet Runner Cookbook.
3. Examine, compile, and run the relevant code samples or snippets for your platform (located in the /SAMPLES/ directory).

4. On all platforms, read the source code comments before compiling and running the sample so that you understand the demonstrated functionality.

Directory structure

The following is the directory structure for all platforms:

```
/AdobePDFLSDK21.0
  /Docs
  /Include
    /Headers
    /Source
  /Libs/PLATFORM
  /Plugins
    /Flattener
      /Include
      /Libs/PLATFORM
      /Samples
    /PDFProcessor
      /Include
      /Libs/PLATFORM
      /Samples
    /XPS2PDF
      /Include
      /Libs/PLATFORM
      /Resource
      /Samples
  /Resource
  /Samples/SAMPLE_NAME
```

Path modification

Because some samples require font, CMap, unicode, or color profile files, modify the MyPDFLibUtils.cpp file with the folder paths on your system.

For example, `PDFLInitHFT()`'s function calls `PDFLGetDirList`, `PDFLGetCMapDir`, `PDFLGetUnicodeDir`, and `PDFLGetColorProfileDirList` to locate the font, CMap, unicode, color profile files on your system when they pass PDFLDataRec's `dirList`, `cMapDirectory`, `unicodeDirectory`, and `colorProfileDirList` members.

Best practices

General

- Examine, compile, and run the relevant code samples or snippets for your platform. The code samples and the SnippetRunner reside in the AdobePDFLSDK21.0/samples directory.
- On all platforms, read the comments in the source code before compiling and running the sample to understand the functionality demonstrated by the sample.

Required best practices

1. All PDFLSDK API calls should go from Host Function Table (HFT). To enable HFT, all projects should be initialized with PDFLInitHFT and should be terminated with PDFLTermHFT APIs respectively. Host Function Table (HFT) has been in place since PDFL 7.0 and all new APIs added after 7.0 are accessible through HFT only.
2. From 18.0 onwards, all PDFLSDK client projects should enable C++ based Exception Handling. To achieve that kindly set `USE_CPLUSPLUS_EXCEPTIONS_FOR_ASEXCEPTIONS` flag to 1.
3. To enable HFT and C++ based Exception Handling efficiently, please include the platform specific project-setting files available in Samples/utils folder.
4. Samples are tested and built with the valid values in the PDFLDataRecstructure. If you do not have the valid font, CMap, unicode, or color profile folders on your system, samples may crash or throw an exception. You must either install the font and color profile files in the correct folder or you must pass NULL to these members for the PDF library not to look for the fonts or color profile files.

Windows installation

The following two archives are shipped for Microsoft Windows:

- AdobePDFLSDKMinSize21.0.zip - For standard version of the PDF Library SDK.
- AdobePDFLSDKMaxSpeed21.0.zip - For the speed-optimized version of the PDF Library SDK.

1. Verify the signature of the ZIP file:

1. Install JDK (preferably JDK 1.6 or higher) on your machine.
2. From the bin folder of the JDK installation, execute the following command:

```
jarsigner.exe -verify [-verbose -certs] %ZIP_File_Path%
```

Note

The following parameters are optional: `-verbose`, `-certs`

2. Unzip the preferred ZIP file: **Important:** If both the above variants need to exist on the disk, extract them to separate locations.
 - For standard version of the PDF Library SDK, use AdobePDFLSDKMinSize21.0.zip.
 - For the speed-optimized version of the PDF Library SDK, use AdobePDFLSDKMaxSpeed21.0.zip.
3. To ensure that applications run successfully outside the Microsoft Visual Studio C++ .NET IDE, verify the executable can find the dynamic linked libraries at run time:
 1. Copy the libraries supplied in the Libs folder to the folder where your executable resides.
 2. Place the path for the libraries into the PATH environment variable.
4. To ensure that the plugins run successfully:
 1. Copy the libraries supplied in the Plugins[PLUGIN_NAME]Libs folder to the folder where your executable resides.

2. Place the path for the libraries into the PATH environment variable.
5. Include the PDFLInitCommon.cpp file available in Include/Source folder to enable C++ Exception Handling across multiple HFT calls.

Note: Executing SnippetRunner in UI mode (64-bit): You must explicitly set the **64BitMode** parameter to 1 in the pdfsdk.config file in your home directory. This variable is initialized to 0 (zero) by default.

Mac installation

The DMG file, AdobePDFLSDK21.0.dmg, is shipped for the Mac platform. To install:

1. Verify the AdobePDFLSDK21.0.dmg. From 18.0.2 onwards, the signing process on Mac platform has been changed. Now, the SIG file is not required. To view the signature of the DMG file, you can use the following codesign command:

```
codesign -dvv $DMG_File_Path.
```

```
codesign -dvv AdobePDFLSDK21.0.dmg
```

Note

The signature information appears. If it is unsigned, "code object is not signed at all" appears.

2. Mount the DMG image for AdobePDFLSDK21.0.dmg.
3. Copy the AdobePDFLSDK21.0 folder to a local drive.

By default, all libs are present under the AdobePDFLSDK21.0/Libs folder; categorized under mac_x64 for 64-bit target configuration. Aliases or proper paths must be set to ensure that the sample applications run properly.

Linux installation

1. Copy the file AdobePDFLSDK21.0.tar.gz to a directory.
2. Run the following command in the directory that you want to unpack the SDK:

```
gunzip < AdobePDFLSDK21.0.tar.gz | tar xvf
```

3. Change directories to AdobePDFLSDK21.0/Samples/utils and modify the appropriate .mak file to point to your installed Clang directory and static library directories.
4. Set the environment variable LD_LIBRARY_PATH to point to the AdobePDFLSDK21.0/Libs/linux_x64/ folder

Before you can build your product on a Linux platform running the Ubuntu 18.0.4 operating system, install the Clang compiler. We recommend using version Clang 9 and keeping the compiler in the directory: /opt/llvm/

Additional documentation

- [PDF Reference, sixth edition, version 1.7](#)
- PDF Redaction: Addendum to the PDF Reference, sixth edition, version 1.7
- Errata for the PDF Reference, sixth edition, version
- Adobe Supplement to the ISO 32000, BaseVersion 1.7, ExtensionLevel 3 (for Acrobat XI)
- [PDF Library API Reference](#). The API Reference is the reference manual for all APIs exposed by the PDF Library.
- [PDF Library Overview](#): Provides an introduction to development using the Adobe PDF Library.

3rd party software notices and TOU

Copyright 2024 Adobe Systems Incorporated and its licensors. All rights reserved. Adobe and the Adobe logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

This product contains either BSAFE and/or TIPEM software by RSA Security, Inc. Portions utilize Microsoft Windows Media Technologies. Copyright (c) 2006 Microsoft Corporation. All Rights Reserved.

Notices, terms and conditions pertaining to other third party software are located at <https://www.adobe.com/go/thirdparty> and incorporated herein by reference.