

PDFTron DocPub User Manual

Version 1.1

1. Introduction

1.1 An Introduction to PDFTron DocPub

PDFTron's DocPub is an easy-to-use, multi-platform executable that provides users with a high-quality and efficient way to convert wide range of documents to PDF, XPS, XOD, HTML and EPUB formats.

Like other PDFTron products, DocPub does not rely on any third party components. The technology is suitable for use in high-throughput server environments.

1.1.1 Key Functions

- Fast, high-quality conversion from practically any document to PDF (Portable Document Format), XPS (XML Paper Specification) or XOD (web optimized OpenXPS standard), HTML or EPUB format, that maintains the original document quality and layout and preserves hyperlinks, colors and fonts.
- Create XOD documents for use with PDFTron WebViewer (HTML5, Silverlight, and Flash), including streaming conversions on-demand.
- The conversion process preserves the original document's meta-data as well as other non-graphical information such as bookmarks, logical structure, and articles.
- Thumbnail generation option for fast navigation through multi-page documents.
- Automatic repair of broken PDF documents.
- Support for all versions of the PDF Language Standard, including Acrobat 9 documents as well as ISO PDF (ISO 32000).
- Batch conversion.
- Wild card and subfolder processing.
- 100% conversion accuracy.
- Optional flattening to create PDF's that render faster on mobile devices with lower memory and speeds.

1.1.2 Common Use Case Scenarios

- Developers may want to use DocPub to quickly add batch conversion support to any application or workflow that requires PDF, XPS, XOD, HTML or EPUB as output formats.
- Server-based, on-demand conversion of documents to PDF, XPS, XOD, HTML and EPUB files.
- Stream XOD conversions on-demand to a WebViewer client.
- Batch processing of large collections of files with the same conversion options.
- Extending existing applications to take advantage of the new XPS Print API and XPS print path available in Windows 8, 7, and Vista.

1.1.3 Operating Systems Supported

- Windows, Linux and Mac.

1.1.4 System Requirements

- At least 12 MB of free disk space.
- Memory requirement is dependent on source document being converted.

3.4 General Usage Examples

Example 1. The simplest command line: Convert a file to PDF or XOD format.

Notes:

- Converts 'my.doc' to 'my.pdf' located in the current working folder.

```
DocPub my.doc
```

- Converts 'my.doc' to 'my.xod' located in the current working folder.

```
DocPub -f xod my.doc
```

Example 2. Convert XOD file to PDF.

Notes:

- The '-o' (or --output) parameter is used to specify the output folder. If this option was not specified, all images would be stored in the current working folder.
- The -f pdf specifies that the output should be an PDF file.
- The '--verb' option instructs DocPub to output more feedback in the console window.

```
DocPub -f pdf --verb 2 -o my_output_folder my.xod
```

Example 3. Batch convert all files to XOD.

Notes:

- The -f xod option instructs DocPub to convert all files in dir1 folder and its subfolders to XPS format.
- The '--subfolders' option is used to recursively process all PDF documents stored in subfolders of dir1 and dir2.

```
DocPub -f xod --subfolders dir1 dir2
```

3.5 Batch Processing and the Use of Wildcards

DocPub supports processing of multiple input documents in the same run. For example, it is possible to specify multiple PDF folders and DocPub will automatically process all documents matching a given file extension. For example, the following command-line will process all PDF documents in folders 'test1' and 'test2'

```
c:\> DocPub --extension pdf -o c:/output_folder c:/test1 c:/test2
```

Wildcard characters can also be used to process multiple input files.

For example, if a directory contains the following PDF documents:

```

C:\test1 >dir
Directory of C:\test1
01/04/2007  03:35 PM    <DIR>          .
01/04/2007  03:35 PM    <DIR>          ..

```

```
05/21/2004 02:27 PM A1.pdf
05/03/2005 09:38 AM A2.pdf
05/20/2003 08:46 AM B1.pdf
05/15/2003 12:50 PM B2.pdf
```

To process all PDF documents in this folder and convert them to XOD format, you could specify:

```
c:\>DocPub -f xod -o c:/output_folder c:/test1/*.pdf
```

To process all PDF documents starting with 'A', you could specify:

```
DocPub -f xod -o c:/output_folder c:/test1/A*.pdf
```

Or to process all PDF documents ending with '1' and convert them to XOD documents, you could specify:

```
DocPub -f xod -o c:/output_folder c:/test1/*1.pdf
```

You can use either of the two standard wildcards — the question mark (?) and the asterisk (*) — to specify filename and path arguments on the command line.

The wildcards are expanded in the same manner as operating system commands. (Please refer to your operating system user's guide if you are unfamiliar with wildcards). Enclosing an argument in double quotation marks (" ") suppresses the wildcard expansion. Within quoted arguments, you can represent quotation marks literally by preceding the double-quotation-mark character with a backslash (\). If no matches are found for the wildcard argument, the argument is passed literally.

3.6 Exit Codes

To provide additional feedback, DocPub returns exit codes after completing processing. The exit codes can be used to provide user feedback, for logging etc. This is particularly important for applications running in an unattended environment.

The following table lists possible exit codes and their description:

Exit Code	Description
0	All files converted successfully
1	Unspecified error
2	Bad license key
3	Failed to create or write the output file or directory
4	Failed to connect to server

All codes other than '0' indicate that there was an error during the conversion process.

To get detailed information on an error, set the `--verb` parameter to 2.

The following illustrates a sample Windows batch script that processes exit codes:

```
@echo off
rem convert all files in 'data' folder

DocPub ./data
if errorlevel 1 goto other_error
if errorlevel 3 goto failed_create_err
if errorlevel 0 goto exit
```

```
:failed_create_err
echo Failed to create a directory.
goto exit

:othererror
echo An error encountered during processing.
goto exit

:exit
```

4. Frequently Asked Questions

4.1 General FAQ

4.1.1 Is DocPub available as a toolkit (SDK) for integration with third party applications?

For developers who are looking for more control over the conversion process than a command-line utility can provide, developers can also license DocPub functionality as part of PDFNet SDK and the applicable add-ons for the required conversions (such as WebViewer Add-on for conversion to XOD, or the PDF to HTML Add-on, etc.). PDFNet SDK is an industry-leading, high-quality document core technology platform powering mobile, server, desktop, web, and cloud-based apps. Using the PDFNet library, developers can create powerful PDF document solutions and applications that can convert, generate, manipulate, optimize, print, view, and markup PDF documents without any third-party software dependencies.

PDFNet SDK is available as a .NET component, and as a cross-platform C/C++, Java, Ruby, Python, PHP and Objective-C, PDF library, that is available on a wide range of platforms (i.e. Windows, Linux, Mac OS X, Android, iOS, Windows 8/RT, etc).

Alternatively, developers can also subscribe to [PDFTron Web Services \(PWS\)](#), a pay-as-you-go platform for licensing the DocPub, WebViewer or PDFNet conversion capabilities. PWS enables developers to sign up to a Cloud account, with the document conversion hosted by PDFTron, or to an On-Premise account with the conversion processing performed on their own servers.

For more details, please visit PDFTron's website at <http://www.pdftron.com> or contact a PDFTron representative via info@pdftron.com or sales@pdftron.com.

4.1.2 What is XPS?

"XPS" stands for "XML Paper Specification" and is a new document format as well as the native print spooler format in Microsoft's Windows Vista® and Windows® 7. The XPS document format consists of XML markup that defines the layout of a document and the visual appearance of each page along with rendering rules for distributing, archiving, rendering, processing and printing the documents. Just like PDF, the XPS document format enables users to view, print, and archive any type of documents without the original program that created them and without loss of fidelity.

To find out more about XPS, please visit Microsoft's website at: <http://www.microsoft.com/whdc/xps/default.mspx>

viewers are typically better at handling this type of issues. The aliasing artifacts should not be visible during printing or high-resolution output.

4.2.3 Why are some pages rasterized?

Occasionally, there will be documents that don't allow for an accurate conversion. In these cases DocPub will by default render the necessary elements at 150 dpi or for XOD documents at the resolution given by the '--dpi' parameter. In other select cases, to generate the exact appearance of the original document, an entire page may need to be rendered. In these cases rendering can be disabled using the '--norender' option. Disabling rendering will preserve resolution independent properties of input PDF documents (including fonts, paths, and shadings) as well as text selection and extraction capability in XPS processing software. Note that when flattening option is enabled that content may still get rasterized, but for different reasons.

4.2.4 Why are some fonts in PDF not rendered consistently?

PDF format, unlike XOD, does not require mandatory font embedding. As a result PDF consumers, such as DocPub and your favorite PDF viewer, need to find substitute fonts for missing fonts on the client system. Unfortunately, this means that there is no guarantee that file will render accurately on different systems or even in different PDF viewers. Default font substitution can be overridden using PDFNet SDK which offers additional options that are not available in the DocPub Command-Line Utility. To avoid font substitution errors, simply make sure to create PDF documents with all fonts embedded. If you cannot guarantee embedded fonts, then make sure the converting computer has the required, or similar, fonts installed. DocPub will do its best to match missing fonts with existing ones on the system.

4.2.5 Why is the image quality so low?

To prevent file size from getting too large there is a maximum size for images. This also allows the generated document to be viewed on low resource devices where there are megapixel limits. You can tell that this limit has been reached when increasing the DPI has no effect. You can therefore increase the --max_image_pixels to a value of say 3 or 5 million, which should suffice. If you don't want any limit then put in a very large number.

4.2.6 Why does the Silverlight plugin crash?

Unfortunately there is a long standing bug in Silverlight that causes it to crash when there are rotated text/glyphs. If you are encountering this issue, then use the --silverlight_text_workaround flag. This has the downside of the text no longer being selectable.

4.2.7 Why do shadows (masks) not appear correctly in Silverlight?

Silverlight does not provide full support for opacity masks as specified in the XPS format. If the generated output does not appear correct in Silverlight then set the --mask_workaround flag, which will detect and then rasterize these cases so they appear correctly, but no longer support zooming.

4.2.8 How do I stream XOD conversions?

To stream the generated XOD file as it is converted simply set the --console_out flag. With this flag the output is sent to the standard out pipe, so you can redirect the pipe to anything you like.

