Datalogics

Datalogics FLIP2PDF™

User Guide

FLIP2PDF version 2.0

For additional information, contact:

Datalogics, Incorporated
www.datalogics.com
Phone: (312) 853-8200

Table of Contents

Introduction

What is FLIP2PDF?

FLIP2PDF is an application that converts disparate files and web content into standard PDF documents. With FLIP2PDF you can create uniform output content, allowing you to deliver a consistent customer experience.

FLIP2PDF is available for Linux and Windows 64-bit platforms, and you can install the software on local hardware or on a cloud computing server to configure and run automated or ad-hoc conversion requests using your command line interface. You can also make conversion requests through FLIP2PDF's API layer. FLIP2PDF supports conversion to PDF from the following file and content types:

- HTML files and web pages
- PostScript and Encapsulated PostScript
- Word documents, doc and docx
- Excel spreadsheets, xls and xlsx
- PowerPoint documents, ppt and pptx
- TIFF, JPEG, BMP, and PNG graphic files

What you get when you buy FLIP2PDF

- An EXE installation file for Windows 64 or BSX for Linux 64
- A command line utility for converting documents into PDF files
- A C language API
- An API sample program you can experiment with out of the box or use as a template for your own development
- Online access to user documentation on our digital Developer Resources site.
- Technical support from our team of digital document specialists and professionals. You can contact your Datalogics Support representative directly by electronic mail or visit our support site for this product.
- Regular software updates

NOTE: Make sure that any HTML or JavaScript files you attempt to process using FLIP2PDF come from a reliable source, or are scanned to make sure that they do not harbor any viruses or malware.

Some Notes on Installing and Running FLIP2PDF

If you purchase the Linux version of the product, please make sure that you install the following before you install FLIP2PDF:

- the Java Runtime Environment (JRE)
- the libcanberra-gtk-module library
- the libXinerama library

To run FLIP2PDF on Linux, use Redhat Enterprise Linux 7/CentOS 7.

FLIP2PDF and LibreOffice

Note that for Windows systems, FLIP2PDF and LibreOffice cannot be installed and running separately on the same server platform. The Windows versions of these two products are not compatible. This does not apply to Linux systems. To work properly, LibreOffice needs to be embedded as part of the FLIP2PDF installation instead.

Make sure, therefore, that you install FLIP2PDF on a server or on a Virtual Machine (VM) where LibreOffice is not already running.

If you decide to install FLIP2PDF on a server where LibreOffice is already running, you will need to uninstall LibreOffice first, and remove UNO_PATH from the environment variables from either Windows or Linux. In that event you will also need to preserve any custom templates or settings that you created for LibreOffice.

When you install FLIP2PDF, our software will install a version of LibreOffice as part of the broader installation process. At the beginning of the installation process, if the installer detects a copy of LibreOffice on the same system, you will be prompted to remove LibreOffice, and then the installation process will end with no changes being applied to your system.

Environment Variables

To install FLIP2PDF run the executable for Windows (such as setup_FLIP2PDF_Pro_Win64_2.0.0.exe) or the self-extracting script file for Linux (such as setup_FLIP2PDF_Pro_Linux64_2.0.0.bsx).

For either platform, when you install the software the executable identifies the product as Pro or Premium, and the version number of the release. The Windows installation program for FLIP2PDF adds the location of the FLIP2PDF executable to the PATH in the Windows Environment Variables, so you can run “flip2pdf.exe” from anywhere. For Linux, if you want to run the executable from anywhere, you need to manually add the location of the FLIP2PDF executable to the PATH variable.

The License and the Activation Key

When you first evaluate or purchase FLIP2PDF, you will receive an activation key, and you will be prompted to enter this key when you install the software. In response the system generates a license file for you, flip2pdf_pro.lic or flip2pdf_premium.lic, and stores it in the software installation directory.

If you later receive a new activation key for FLIP2PDF, you will need to enter it as you did your original activation key when you first installed the product. Delete the license file from the installation directory and then run the executable, flip2pdf.exe or flip2pdf.bsx. You will be prompted to enter the activation key. Type or paste the key value as you did before. The system will create a new license file for you, and you can continue to use the product.

Note that if you don’t enter the activation key value when you first install the product, or enter it incorrectly, you will be prompted to enter the value again the next time you run FLIP2PDF. But if you enter

the key later, the system will create the license file in the working directory where the executable is run. In that event you will need to move the license file from the working directory to the installation directory.

Working with Fonts

By default FLIP2PDF automatically embeds in output PDF documents any fonts used to create those documents. This is considered a best practice for creating PDF files. With the fonts embedded in a PDF file, that file will always look the same wherever it is opened, because the viewer tool (such as Adobe Reader) does not need to look for a font on the local machine. But to generate the PDF document with the embedded font using FLIP2PDF, the software must be able to find it.

Any font used in a PDF document generated by FLIP2PDF must be installed on the machine running FLIP2PDF. Otherwise FLIP2PDF will look for an alternate font file that can be applied to the PDF document as a substitute. If you are using a standard font, like Times New Roman in English, the font is probably on the machine already, installed with the Windows operating system or with Microsoft Office. But you may need to manually install a specialized font on the machine, such as a font with Arabic, Mandarin, or Korean characters, if you seek to use FLIP2PDF to create a PDF document based on a source file using those characters.

Keep in mind that you might create a Word document with Arabic characters on one machine and then transfer that file to another for processing with FLIP2PDF. For example, you might edit a document with Arabic characters in Word and then copy it to a directory on a network server, so that it can be included in an automatic overnight batch process to convert the files in that directory to PDF documents using FLIP2PDF. If that Arabic font is not installed on the server running FLIP2PDF, the Arabic characters will not appear in the PDF output document.

It is also possible to embed a specialized font in a Microsoft Office source document, such as a Word file. Then FLIP2PDF can find the font in the source file itself, rather than on the machine.

Command Syntax

Required and optional values

Every command statement you enter for FLIP2PDF must include the following values:

1. the executable name
2. the name of the input file, or a web site address if you want to convert a web page to a PDF file
3. the name you want to assign to the output PDF file

You can also provide the name of the JSON profile file but that is optional.

```
flip2pdf --input inputname --output outputfilename --profile profilename
```

The JSON profile file includes a list of settings that define exactly what kind of PDF output document you want to produce. See "Setting up your Profile" on page 9.

Two optional values are also available:

- `--password`. If an input document has a password, you can include this value to provide a password so that FLIP2PDF can read and open that file. This only works with Office input files.
- `--encrypt`. Use this value to assign a password to a PDF output document. This only works with Office input files.
- `--joboptions`. Use this value to add the use of a JobOptions file during conversion. This only works with Postscript input files.

For each command line option, you can use either the short or long notation:

- `-i` [`--input`]
- `-o` [`--output`]
- `-j` [`--profile`]
- `-p` [`--password`]
- `-e` [`--encrypt`]
- `-b` [`--joboptions`]

Type `flip2pdf –-help` at the command line for a list of software options.

What a basic command looks like

A command might look like this:

```
flip2pdf --input TestDocument.docx --output TestDocument.pdf --profile
standard.json
```

If you are converting an input file that is password protected, you can supply the password to open that file using a command syntax like this:

```
flip2pdf --input TestDocument.docx --output TestDocument.pdf --profile
standard.json --password Fischer@27
```

Then, if you want to assign the same password ("Fischer@27") to the PDF output file, use this command syntax to supply both passwords:

```
flip2pdf --input TestDocument.docx --output TestDocument.pdf --profile
standard.json --password Fischer@27 --encrypt Fischer@27
```

The FLIP2PDF software assumes that the file extension you provide for an input file is valid.

Providing path names

You don't need to include the absolute path for any of these files if the input file and profile are stored in the current working directory, and if you save the output file to this directory as well.

If you want FLIP2PDF to take an input file from one directory and save the output to another, however, you need to provide the path as well as the file name:

```
flip2pdf --input C:\Datalogics\Source\AnnualReport2016.docx -–output
C:\Datalogics\OutputFiles\AnnualReport2016-B.pdf -–profile standard.json
```

If any of the file or path names include spaces, use quotes around the name:

```
flip2pdf --input "C:\Datalogics\Source\Annual Report 2016.docx" -–output
C:\Datalogics\OutputFiles\AnnualReport2016-B.pdf -–profile standard.json
```

Converting web page content to PDF

You can also convert an HTML file or a web page to PDF output. Enter the name or the name and the path for that HTML file in the command line statement, as you would any other kind of file, like this:

```
flip2pdf --input C:\Datalogics\Source\AnnualReport2016.html --output
C:\Datalogics\OutputFiles\AnnualReport2016-B.pdf --profile standard.json
```

If you want to convert a web page to a PDF file, enter the web site address in the command line statement instead.

```
flip2pdf --input https://www.datalogics.com/products/pdftools/flip2pdf/ --
output C:\Datalogics\OutputFiles\FLIP2PDF_Page.pdf -–profile standard.json
```

The product will convert all of the content shown on that page to a PDF output file, including HTML, stylesheets, and graphics.

Using the JSON profile

The JSON profile file is not required. If you are content to use the default settings provided with the FLIP2PDF software, you can enter a command statement like this:

```
flip2pdf --input TestDocument.docx --output TestDocument.pdf
```

If you do specify a JSON profile file name in your command statement, you can use either the relative or the absolute path for that file. If you don't provide an absolute path for the JSON profile, FLIP2PDF will look for the file in the working directory. If the file is not in the working directory, FLIP2PDF will look for the JSON profile in the FLIP2PDFProfiles directory where the product was installed.

If FLIP2PDF can't find the JSON profile, you will see an error message.

These are the FLIP2PDF default settings:

output	PDF file
compression method	jpeg
downsampling	All images are downsampled by default to 300 DPI
tagged PDF	Do not create a tagged PDF document when creating an Office file

In other words, if you don't include a JSON profile in your command line statement FLIP2PDF will generate a standard PDF document. The file will not be tagged, and any graphics found in a PostScript or Microsoft source document or on a web page will be downsampled to 300 DPI and compressed using the jpeg compression method. Also, the fonts used by the PDF output file will be embedded in that PDF file.

Conversion Options

Using FLIP2PDF, you can select the following options for the PDF output in your JSON profile file:

pdf-output-type	Create a **PDF, PDF/A, or PDF/X** document.	**Microsoft Office:** PDF, PDF/A-1 **PostScript:** PDF, PDF/X-3
compression	Select either **lossy (JPEG) or lossless (Flate/ZIP)** compression for images embedded in PostScript or Microsoft Office files, or found on web pages for HTML conversion.	**Microsoft Office:** jpeg, flate **PostScript:** jpeg, flate **HTML:** jpeg, flate

downsample	**Downsample** images from the input files or preserve the original resolutions for these images.	**Microsoft Office:** on, off **PostScript:** on, off **HTML:** on, off
downsampling-dpi	Set the resolution for **downsampling** images in **dots per inch**.	**Microsoft Office:** 75, 150, 300, 600, 1200 **PostScript:** 75, 150, 300, 600, 1200 **HTML:** 75, 150, 300, 600, 1200
rotate-pages	Control the interpretation of the PostScript "ViewingOrientation" parameter.	**PostScript:** all, none
tagged-pdf	Create a **tagged PDF document** for use in providing accessibility to disabled users.	**Microsoft Office:** on, off
page-size	Select a **page size** for the PDF file.	**HTML:** letter, legal, ledger, A3, A4, A5
page-margin	Set **margins** for a PDF file in inches (in) or millimeters (mm).	**HTML:** "0.75in", "15mm"
page-orientation	Set the page orientation to **portrait** or **landscape.**	**HTML:** portrait, landscape
web-layout	For web pages with responsive design, select the **web layout** intended **desktop**, **tablet**, or **mobile** viewing.	**HTML:** desktop, tablet, mobile

PDF/A and PDF/X

For PostScript files, select either PDF or PDF/X as output in the JSON profile.

For Office files, select either PDF or PDF/A as output in the JSON profile. FLIP2PDF can generate PDF/A-1a output files.

Note: FLIP2PDF can generate PDF/A archive files but the software cannot certify that these files satisfy ISO standards for PDF/A conformance.

Lossy or Lossless Compression

Lossy and lossless refer to the approach used for compressing data. For lossless, all of the data in the image is preserved. The quality of the image does not change, and it can be uncompressed to its original state. Lossy compression permanently removes data from the image file, such as pixels, reducing the image resolution. Files reduced using lossy compression will be considerably smaller but will not print or display as well as those compressed using lossless compression.

For lossless compression, you would enter this statement in the JSON profile:

```
"compression" : "flate"
```

For lossy, use this value instead:

```
"compression" : "jpeg"
```

Downsampling

Downsampling involves reducing the size of a graphic file when generating output. If you know that you do not need high resolution images embedded in a PDF document, and you are working with a Microsoft Office or PostScript source file, you can downsample these images to a lower resolution to reduce the size of the PDF document itself. The downsampling process involves changing the width and height of an image in pixels to reach a given target resolution that FLIP2PDF allows you to select. Target options for downsampling images include 75, 150, 300, 600, and 1200 Dots per Inch (DPI).

FLIP2PDF is set up to downsample images by default, and each one to 300 DPI. If you set the downsample option to "off" the process will preserve the original resolution of each image in the PDF output file. Or you can set downsample to "on" and then select your own target resolution, greater or less than 300 DPI. The software preserves any design or size information found in an image.

Tagged PDF Document

For Microsoft Office files, you can choose to produce a tagged PDF. A tagged PDF document contains metadata to describe instructions related to headers and other content on a page. Tagging is generally used with a PDF document to meet accessibility requirements. For example, tags in a PDF document might be placed so that text, headings, footnotes and other content in the document can be interpreted by a screen reading software tool. The tool could use this information to read text in the document out loud for a blind person or respond to voice commands for a reader who can't easily use a mouse or keyboard.

Margins, page size, and page orientation

For HTML files and web pages, you can set options to define a margin around the output PDF document pages, choose a standard PDF page size, or set those pages to landscape orientation rather than portrait.

PostScript files contain information on how pages should be oriented. You can control whether this information will be respected or ignored for all pages.

Select the web layout for responsive web pages

For web pages designed to be responsive to different device screen sizes, you can choose the layout intended for desktop, tablet, or mobile viewing. This will not affect the size of the output PDF but may result in different layout of the content, depending on whether the HTML input supports responsive design.

Setting up your Profile

Your JSON profile file (or files) should include a list of settings that define exactly what kind of PDF output document you want to produce. We provide a default JSON file called “standard.json.” You can edit this JSON profile as you like or use it as a template for creating your own.

Every setting for FLIP2PDF is optional. That means then that a setting is only applied if it is included in the JSON profile. Flag settings must be set to “on” to be activated. Settings that are turned off do not need to be defined in the JSON profile file.

You can use the JSON validator JSONLint to check your JSON syntax (https://jsonlint.com).

The standard.json Profile

This is what the standard.json profile looks like, with all options provided.

```
{
    "postscript": {
        "color-images": {
          "downsample" : "on",
          "downsampling-dpi" : 300,
          "compression" : "jpeg"
        },
        "gray-images": {
          "downsample" : "on",
          "downsampling-dpi" : 300,
          "compression" : "jpeg"
        },
        "mono-images": {
          "downsample" : "on",
          "downsampling-dpi" : 300
        },
        "rotate-pages" : "all",
        "pdf-output-type" : "PDF"
    },
    "office": {
        "images": {
          "downsample" : "on",
          "downsampling-dpi" : 300,
          "compression" : "jpeg"
        },
        "tagged-pdf" : "off",
        "pdf-output-type" : "PDF"
    },
    "html": {
        "images": {
```

```
            "downsample" : "on",
            "downsampling-dpi" : 300,
            "compression" : "jpeg"
        },
        "web-layout" : "desktop",
        "page-size" : "letter",
        "page-margin" : "1.0in",
        "page-orientation" : "portrait"
    }
}
```

Converting PostScript Files

JSON profile parameters:

downsample	Set this value to “on” if you want to enable the ability to select a resolution value for downsampling graphics images in a PostScript file.	
	downsampling-dpi	Enter a resolution value, in Dots per Inch, if you want to reduce the resolution of the graphics embedded in a PostScript file. This may result in a smaller PDF output document. The available options are 75, 150, 300, 600, and 1200. Defaults to 300 DPI.
compression	Enter a compression method to use for graphics images in PostScript files. Enter jpeg or flate. Defaults to jpeg.	
rotate-pages	Set this value to “all” to respect the “ViewingOrientation” setting in the PostScript file. Set this value to “none” to ignore the setting. This will affect the orientation of the pages in the PDF output.	
pdf-output-type	Enter an output type, PDF, or use PDF/X-3 for graphics use. Defaults to PDF.	

If you wanted to convert a PostScript file with embedded color images, reduce the resolution for these images to 600 DPI while using the default “jpeg” compression method, and save the PDF output documents as PDF/X files, the PostScript section of your JSON profile might look like this:

```
{
    "postscript": {
        "color-images": {
            "downsample" : "on",
            "downsampling-dpi" : 600,
        }
        "pdf-output-type" : "PDF/X-3"
    }
}
```

If you weren't planning to convert any PostScript files into PDF documents, you could leave the PostScript parameters out of your JSON profile.

Converting Microsoft Office Files

JSON profile parameters:

downsample	Set this value to "on" if you want to enable the ability to select a resolution value for downsampling graphics images in a Microsoft Office file.	
	downsampling-dpi	Enter a resolution value, in Dots per Inch, if you want to reduce the resolution of the graphics embedded in an Office file. This may result in a smaller PDF output document. The available options are 75, 150, 300, 600, and 1200. Defaults to 300 DPI.
compression	Enter a compression method to use for color or grayscale graphics images in Office files. Enter jpeg or flate. Defaults to jpeg.	
tagged-pdf	Set to "on" to create a tagged PDF document. The tagged PDF feature only works with Office documents. You could add tags to a Microsoft Word document and then enable the tagged-pdf option, and FLIP2PDF would create a tagged PDF in response. Defaults to off. Note that if you select PDF/A as the pdf-output-type parameter, the software will generate a tagged PDF document as the output file. It will ignore the setting for the tagged-pdf parameter.	
pdf-output-type	Enter an output type, PDF or PDF/A-1. Defaults to PDF.	

You might, for example, want to convert a folder containing Microsoft Word documents with embedded images and downsample those images to 300 DPI using the JPEG compression method while saving the output files as PDF/A output files. You don't need to create tagged PDF documents. The Microsoft Office section of your JSON profile might look like this:

```
{
    "office": {
        "images": {
          "downsample" : "on",
          "downsampling-dpi" : 600,
          "compression" : "jpeg"
        },
        "pdf-output-type" : "PDF/A-1"
    }
}
```

If you weren't planning to convert any Office files into PDF documents, you could leave the Office parameters out of your JSON profile.

Converting HTML Files and Web Site Pages

JSON profile parameters:

downsample	Set this value to "on" if you want to enable the ability to select a resolution value for downsampling graphics images found on a web site.
downsampling-dpi	Enter a resolution value, in Dots per Inch, if you want to reduce the resolution of the graphics found on a web site. This may result in a smaller PDF output document. The available options are 75, 150, 300, 600, and 1200. Defaults to 300 DPI.
compression	Enter a compression method to use for graphics images found on a web site. Enter jpeg or flate. Defaults to jpeg.
web-layout	Select an option for the web layout to use for responsive HTML pages— desktop, tablet, or mobile. Defaults to desktop. In Responsive Web Design, a web page can be designed to appear differently depending on the screen size of the device used to display that web page. It is easier to read and navigate a web site on a mobile phone if that web site is specifically designed to present on a mobile phone. With this option in FLIP2PDF, you can select the layout for responsive pages that is best suited for your needs. Each option provided corresponds to a set of pixel measurements that sets the web viewport. The default for desktop viewing, for example, is 1280 x 1024 pixels.
page-size	Defaults to letter. Enter the size for the page in the PDF output document, letter, legal, ledger, A3, A4, or A5.
page-margin	Defaults to one inch margins on each PDF output page. Enter a number and unit (inches or millimeters) to set a page margin. For example, enter "2.0in" for two inches or "10mm" for ten millimeters.
page-orientation	Defaults to portrait. Set this value to "landscape" to generate landscape PDF documents instead.

You might, for example, want to convert a web page and downsample the images shown on that page to 600 DPI using the JPEG compression method. You could set the page margins to 10 millimeters for a standard A4 page size and use the landscape page orientation, while using the tablet web page layout. The HTML section of your JSON profile might look like this:

```
    }
    "html": {
        "images": {
          "downsample" : "on",
```

```
        "downsampling-dpi" : 600,
        "compression" : "jpeg"
      },
      "web-layout" : "tablet",
      "page-size" : "A4",
      "page-margin" : "10mm",
      "page-orientation" : "landscape"
  }
}
```

If you weren't planning to convert any HTML files or web sites to PDF documents, you could leave the HTML parameters out of your JSON profile.

Converting Graphics Files

Your command line statement for converting graphics files might look like this:

```
flip2pdf --input TestImage.jpg --output TestImage.pdf
```

There are no options to enter in the JSON profile for converting graphics image files.

FLIP2PDF does not downsample graphics files when converting them to PDF documents. The product will use the resolution assigned to each BMP, JPEG, TIFF, or PNG file when creating the PDF output. Sometimes, however, an image file does not have resolution information saved with that file. If an image does not have a defined resolution, FLIP2PDF writes it to 72 DPI when generating the output PDF document.

FLIP2PDF uses the flate compression method when converting BMP, PNG, and TIF graphics files to PDF documents. For JPG files, the product uses jpeg compression.

FLIP2PDF preserves transparencies in PNG and TIFF images in the PDF output and preserves the color space for raster images that are RGB, CMYK, or Grayscale. FLIP2PDF does not process TIFF files that use color spaces other than Grayscale, RGB or CMYK.

Using the FLIP2PDF Software Development Kit (SDK)

FLIP2PDF includes a simple C language API for converting input files into PDF documents for both Windows and Linux platforms.

Note that the structure used by the API header file for FLIP2PDF follows the pattern of the JSON profile file used for the FLIP2PDF command line utility. The FLIP2PDF API declarations are found in the header file FLIP2PDFAPI.h. Under the FLIP2PDFParameters control structure we offer a series of other control structures for PostScript and Microsoft Office files.

The product does not offer any settings for processing image files being converted into PDF output documents, such as JPG, BMP, or PNG files. The API uses the resolution defined in the input graphic file, and assigns a default compression method, jpeg for JPG files and Flate for all other file types.

When you convert HTML to PDF, the PDF output is not tagged, and the output is a standard PDF file by default. It is not possible to convert HTML content to PDF/A.

Parameter Syntax

```
parameters.postscript.colorImages.downsamplingDpi
parameters.postscript.monoImages.downsamplingDpi
parameters.postscript.grayImages.downsamplingDpi
parameters.postscript.colorImages.compression
parameters.postscript.monoImages.compression
parameters.postscript.grayImages.compression
parameters.postscript.rotatePagesType
parameters.postscript.outputType
parameters.postscript.jobOptions

parameters.office.images.downsamplingDpi
parameters.office.images.compression
parameters.office.taggedPdf
parameters.office.outputType
parameters.office.password
parameters.office.encrypt

parameters.html.images.downsamplingDpi
parameters.html.images.compression
parameters.html.webLayout
parameters.html.pageSize
parameters.html.pageMargin
parameters.html.pageOrientation
```

Settings

Parameters	Description	Settings
ImageDownsamplingDpi_t	PostScript files Microsoft Office files HTML files and web pages Select a resolution value in Dots per Inch to reduce the resolution of graphics embedded in the input. The options are 75, 150, 300, 600, and 1200 DPI. The default value is set to 300 DPI. Changing the resolution of images in the input file may result in a smaller PDF output document. To disable the feature to downsample images in a file, use this parameter setting: ImageDownsamplingDpi_Disabled = -1 If downsampling is disabled, FLIP2PDF will copy any images found in an import file directly to the PDF output file without reducing the resolution.	Disabled = -1 300 = 0 (default) 75 150 600 1200
This parameter sets the resolution for downsampling color images in a PostScript input file to 300 DPI: **`parameters.postscript.colorImages.downsamplingDpi = ImageDownsamplingDpi_300;`**		
ImageCompression_t	PostScript files Microsoft Office files HTML files and web pages Select a compression method to use for graphics images in the input file, JPEG or Flate.	JPEG = 0 Flate = 1
This parameter sets the compression method for images in an Office input file to Flate: **`parameters.office.images.compression = ImageCompression_Flate;`**		
TaggedPDF_t	Microsoft Office files only Set to ON to enable. Defaults to OFF. This parameter creates a tagged PDF document for an input file. If you enable TaggedPDF, FLIP2PDF creates a	OFF = 0 ON = 1

	tagged PDF document corresponding to the semantic structures found in the Office source file. Titles, headings, lists and other structures in the document are expressed in the output PDF.	
This parameter creates a tagged PDF output document from an Office file: **parameters.office.taggedPdf = TaggedPDF_ON;**		
RotatePagesType_t	PostScript files only Select whether the PostScript "ViewingOrientation" setting will be respected for all pages, or no pages.	All = 0 None = 1
This parameter creates a PDF output document whose pages ignore the "ViewingOrientation" setting in the PostScript file: **parameters.postscript.RotatePagesType = RotatePagesType_None;**		
PDFOutputType_t	PostScript files Microsoft Office Files Select an output type. Defaults to PDF. PDFA1 (PDF Archive) is only available for Office files. PDFX3 is only available for PostScript files. If you select PDFOutputType_PDFA1, to generate a PDF/A output file, the software will generate a tagged PDF document as the output file. It will ignore the setting for the TaggedPDF parameter.	PDF = 0 PDFA1 PDFX3
This parameter creates a PDF/A output document from an Office file: **parameters.office.outputType = PDFOutputType_PDFA1;**		
Password_t	Microsoft Office files only If an input document has a password, you can include this value to provide a password so that FLIP2PDF can read and open that file.	Password term
This parameter allows the system to open a password-protected input Office file: **parameters.office.password = Password_Fischer@27;**		

Encrypt_t	Microsoft Office files only Use this value to assign a password to a PDF output document.	Password term
This parameter allows the system to assign a password to a PDF output document based on an Office file: **parameters.office.encrypt = Encrypt_Fischer@27;**		
JobOptions_t	Postscript files only Use this value to use a JobOptions file during conversion.	JobOptions term
This parameter allows the system to use a JobOptions file during Postscript conversion to PDF: **parameters.postscript.jobOptions = "../../AJobOptionsFile.joboptions";**		
WebLayout_t	HTML/Web sites only Select a display target for the PDF output document, based on the device used to display the source web page.	desktop tablet mobile
This parameter allows the system to configure the PDF output document based on the device to use to display that output document, in this example, for a tablet: **parameters.html.webLayout = WebLayout_Tablet;**		
PageSize_t	HTML/Web sites only Use this value to define the paper size to use for the PDF output file.	letter legal ledger (11x17) A3 A4 A5
This parameter allows the system to set the page size to A4, the European standard: **parameters.html.pageSize = PageSize_A4;**		
PageMargin_t	HTML/Web sites only Use this value to define page margins in the PDF output file.	Number of inches or millimeters with "in" or "mm"
This parameter directs the system to set the page margins to 10 millimeters, rather than the default of one inch: **parameters.html.pageMargin = PageMargin_10mm;**		
PageOrientation_t	HTML/Web sites only Use this value to set the page orientation to landscape rather than portrait.	portrait or landscape
This parameter allows the system to set the page orientation sideways, to landscape:		

```
parameters.html.pageOrientation = PageOrientation_Landscape;
```

Functions

Parameters	Description	Settings
FLIP2PDFInitialize()	The API must be initialized before use. The API can only be initialized once for a process.	
FLIP2PDFInitParameters()	The FLIP2PDFParameters block must be initialized before the block can be used.	
FLIP2PDFConvert()	Convert the input document into a PDF file	Parameters Path to the input file Path to the PDF file to be generated Path to a URL for a web site
FLIP2PDFShutdown()	This method shuts down and releases resources. It should be called before the program exits.	
DetermineFileTypeFormat()	Define the type of input file.	Office binary file (doc, xls, ppt) = 0 Office Open XML (docx, xlsx, pptx) = 1 PostScript = 2 Encapsulated PostScript = 3 Binary Encapsulated PostScript = 4 BMP = 5 JPG = 6 PNG = 7 TIFF = 8 HTML = 9
IsFilePathPresent()	Determine if an input file or directory exists so that FLIP2PDF can access the input file and process it.	File path not found = 0 File path exists = 1

The FLIP2PDF Sample Program

We provide the Conversion sample program to demonstrate how to work with the FLIP2PDF API to convert documents to PDF files. You can use this code as a model for creating your own programs, or simply copy and paste this sample code to use as your own. Microsoft Visual Studio 2017 or later is required to build the FLIP2PDF sample program.

This program calls the Initialize and Shutdown functions once, as an example. Every call to Initialize should be matched with a Shutdown call.

The sample allows you to specify the names of the input files and the output PDF document. You can also change settings in this sample to complete a variety of processing steps when generating a PDF document, such as:

- setting up an Office input file to use Flate compression for embedded graphics images
- downsampling images in an Office input file to 150 DPI
- generating a PDF/X-3 output file from a PostScript input file

Important dependencies for building your own application

The sample projects that are included in the installation package are set up to produce the file structure required by FLIP2PDF to execute correctly. When building your own application for the FLIP2PDF library, the following files and folders must be located at the same level as the application executable:

- PS.VM
- superatm.db
- startupNORM.ps
- flip2pdf_pro.lic or flip2pdf_premium.lic (for subscription-based versions only)
- ICCProfiles directory
- Fonts directory
- Resources directory

The FLIP2PDF library file finds the directory where the application executable is stored and looks for these files and folders in that directory. If you have installed the Linux version of FLIP2PDF, when you run the makefile to build the Conversion sample program, it copies the required files and folders to Samples/Conversion, as well as the license file. The Windows solution file will place the Conversion executable at the top level install directory where the dependencies can be found.

Error Codes

1001 The product found a syntax error with the command line options.
1002 The JSON profile file was not found.
1003 There was a syntax error in the JSON profile.
1004 One of the values assigned to a parameter in the JSON profile was not valid.
1005 The command line statement is missing a required argument for the input file.
1006 The command line statement is missing a required argument for the output file.
1007 The input file was not found.
1008 The input file found has an unsupported file type.
1009 The software could not connect to the Internet to verify the product license.
1010 License activation unknown.
1011 License inactive. Your license may have been disabled by the vendor.
1012 License activation already in use by another user.
1013 License expired.
1014 License failed. Verify that the activation key was entered properly.
1015 There was a problem in setting up Office. Try restarting your system.
1016 There was a problem with converting an Office file. The file may be corrupt or mislabeled.
1017 There was a problem in creating output for an Office file. The file may be corrupt or mislabeled.
1018 There was a problem in reading the Office input file. The input path may be invalid.
1019 An error occurred in creating output for an Office file. The output path may be invalid.
1020 The Resources directory for Office files is missing. Please check your product installation or reinstall the software.
1021 The PostScript ps.vm file is missing. Please check your product installation or reinstall the software.
1022 The PostScript superatm.db file is missing. Please check your product installation or reinstall the software.
1023 The PostScript startupNORM.ps file is missing. Please check your product installation or reinstall the software.
1024 The PostScript ICCProfiles directory is missing.
1025 The PostScript fonts directory is missing.
1026 There was a problem in converting a PostScript file. The file may be corrupt.
1027 There was a problem creating the output for a PostScript file. The file may be corrupt.
1028 Encapsulated PostScript files with a binary preview are not supported.
1029 FLIP2PDF could not create a PDF/X compliant document from the PostScript input file. Please change the PDF output type to PDF.
1030 You tried to start the FLIP2PDF API, but the API was already initialized and running.
1031 The FLIP2PDF API has not yet been initialized. FLIP2PDFInitialize() must be called first.
1032 Log files created after an Office conversion failed are still present and need to be deleted.

When you start FLIP2PDF, the product looks for any error log files left behind by an Office conversion and deletes them if found. These files can keep FLIP2PDF from starting. If FLIP2PDF fails to delete these files properly, you will see this error message.

The log files are found in this directory in Windows:

```
C:\Users\<<username>>\AppData\Roaming\LibreOffice\4\crash
```

And in this directory in Linux:

```
~/.config/libreoffice/4/crash
```

Delete every file you find in this location and then try to start FLIP2PDF again.

1033 Problem converting PDF document to HTML. Try changing to a different web layout, like using “desktop” instead of “tablet.”

1034 The joboptions path specified by the -joboption parameter does not exist.

Acknowledgements

Datalogics FLIP2PDF includes the following software under the following terms:

Adobe PDF Converter

Portions include software under the following terms:

This product includes software developed by the Apache Software Foundation (http://www.apache.org).

The Apache Software License, Version 1.1 Portions Copyright (c) 1998-2000, 1999-2004, 1999 – 2000, 2000 – 2003 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1.Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2.Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3.The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4.The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5.Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright

(c) 1999, International Business Machines, Inc., http://www.ibm.com. For more information on the Apache Software Foundation, please see http://www.apache.org.

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

Portions developed by the University of California, Berkeley.

Portions Copyright (c) 1990 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and

other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Portions © Eastman Kodak Company, 1999 and used under license. All rights reserved. Kodak is a registered trademark of Eastman Kodak Company.

Crypto++ Library

Compilation Copyright (c) 1995-2013 by Wei Dai. All rights reserved.
This copyright applies only to this software distribution package
as a compilation, and does not imply a copyright on any particular
file in the package.

All individual files in this compilation are placed in the public domain by
Wei Dai and other contributors.

I would like to thank the following authors for placing their works into
the public domain:

Joan Daemen - 3way.cpp
Leonard Janke - cast.cpp, seal.cpp

Steve Reid - cast.cpp
Phil Karn - des.cpp
Andrew M. Kuchling - md2.cpp, md4.cpp
Colin Plumb - md5.cpp
Seal Woods - rc6.cpp
Chris Morgan - rijndael.cpp
Paulo Baretto - rijndael.cpp, skipjack.cpp, square.cpp
Richard De Moliner - safer.cpp
Matthew Skala - twofish.cpp
Kevin Springle - camellia.cpp, shacal2.cpp, ttmac.cpp, whrlpool.cpp, ripemd.cpp
Ronny Van Keer - sha3.cpp

The Crypto++ Library (as a compilation) is currently licensed under the Boost Software License 1.0 (http://www.boost.org/users/license.html).

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

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, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LibreOffice

LibreOffice is © copyright 2018 libreoffice.org. This product includes software licensed under the Mozilla Public License (MPL) v2.0. You may download source code for LibreOffice from https://www.libreoffice.org.

Mozilla Public License, version 2.0 1. Definitions 1.1. “Contributor” means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. “Contributor Version” means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor’s Contribution.

1.3. “Contribution” means Covered Software of a particular Contributor.

1.4. “Covered Software” means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. “Incompatible with Secondary Licenses” means that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. “Executable Form” means any form of the work other than Source Code Form.

1.7. “Larger Work” means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. “License” means this document.

1.9. “Licensable” means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. “Modifications” means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. “Patent Claims” of a Contributor means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. “Secondary License” means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. “Source Code Form” means the form of the work preferred for making modifications.

1.14. “You” (or “Your”) means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions 2.1. Grants Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version);

or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities 3.1. Distribution of Source Form All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order,

or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination 5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, you (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party’s negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License 10.1. New Versions Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible with Secondary Licenses If You choose to distribute Source Code Form that is Incompatible with Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, you can obtain one at http://mozilla.org/MPL/2.0.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible with Secondary Licenses" Notice This Source Code Form is "Incompatible with Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library.

Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.