



# PDFTron PDF2SVG™ User Manual

Version 4.x

PDFTron PDF2SVG™ Command-Line Application User Manual  
Part number: PDFTRON-4-PDF2SVGCMD  
Last Updated: February 1, 2010

© 2003-2010 PDFTron Systems, Inc. All Rights Reserved.

All information contained herein is the property of PDFTron Systems, Inc. ("PDFTron"). No part of this publication (whether in hardcopy or electronic form) may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of PDFTron Systems, Inc.

The information in this publication is provided for informational use only, is subject to change without notice, and should not be construed as a commitment by PDFTron. PDFTron assumes no responsibility or liability for any loss or damage that may arise from the use of any information in this publication. The software described in this user manual is furnished under License (enclosed in the software package) and may only be used or copied in accordance with the terms of that License.

PDFTron and the names of PDFTron products referenced herein are either trademarks and/or service marks and/or registered trademarks of PDFTron Systems, Inc. PDFTron, PDFNet SDK, PDF/A Manager, PDF2Image, PDF2SVG, PDF2Text, PDF2XPS, XPSConvert, PDFTron PDFSecure, PDF PageMaster, CosEdit, PDFNet SDK, PDF2Image SDK, PDF2SVG SDK, PDF2Text SDK, PDF2XPS SDK, XPSConvert SDK, PDFSecure SDK, PDF PageMaster SDK and associated Logos are either trademarks and/or service marks and/or registered trademarks of PDFTron Systems, Inc.

Any other brand or product names mentioned in this publication are the registered trademarks or trademarks of their respective holders. Mention of a product in this document does not necessarily imply endorsement of the product.

<b>Legal Statement and Copyright Notice</b>		<b>2</b>
<b>1. Introduction</b>		<b>5</b>
1.1	<b>An Introduction to PDFTron PDF2SVG</b>	<b>5</b>
1.1.1	Key Functions	<b>5</b>
1.1.2	What's New in PDF2SVG?	<b>6</b>
1.1.3	Why SVG?	<b>6</b>
1.1.4	Common Use Case Scenarios	<b>6</b>
1.1.5	Operating Systems Supported	<b>6</b>
1.1.6	System Requirements	<b>6</b>
1.2	<b>About This Manual</b>	<b>7</b>
<b>2. Installing and Uninstalling PDF2SVG</b>		<b>8</b>
2.1	<b>PDF2SVG Installation</b>	<b>8</b>
2.2	<b>Demo Version Installation</b>	<b>8</b>
2.3	<b>Uninstalling PDF2SVG</b>	<b>8</b>
<b>3. Overview</b>		<b>9</b>
3.1	<b>Basic Syntax</b>	<b>9</b>
3.2	<b>Command-Line Summary</b>	<b>10</b>
3.3	<b>Basic Usage</b>	<b>12</b>
3.3.1	How to save converted files in a given folder?	<b>12</b>
3.3.2	How can I control the output names of generated files?	<b>12</b>
3.3.3	How do I create compressed SVG (SVGZ)?	<b>12</b>
3.3.4	How do I produce stand-alone SVG?	<b>12</b>
3.3.5	How do I open a password protected PDF?	<b>13</b>
3.3.6	Why is PDF2SVG generating page thumbnails and the XML summary document?	<b>13</b>
3.3.7	How do I specify which pages to convert?	<b>13</b>
3.3.8	How do I batch-convert files?	<b>14</b>
3.3.9	How can I show/hide crop marks or the trim region?	<b>14</b>
3.4	<b>General Usage Examples</b>	<b>16</b>
<b>Example 1.</b> The simplest command line: Convert PDF to SVG.		<b>16</b>
<b>Example 2.</b> Convert PDF to compressed SVG and without thumbnails and XML summary.		<b>16</b>
<b>Example 3.</b> Convert a password protected file to SVG.		<b>16</b>
<b>Example 4.</b> Convert all PDF document in a given folders to stand alone SVG.		<b>16</b>
3.5	<b>Batch Processing and the Use of Wildcards</b>	<b>17</b>
3.6	<b>Exit Codes</b>	<b>18</b>
3.7	<b>Frequently Asked Questions</b>	<b>19</b>
3.7.1	What is SVG?	<b>19</b>
3.7.2	Why are some characters missing in SVG?	<b>19</b>

4

## 1. Introduction

### 1.1 An Introduction to PDFTron PDF2SVG

PDFTron's PDF2SVG is an efficient, stand-alone command-line application that enables users to convert and publish PDF documents as SVG (Scalable Vector Graphics), the open-standard W3C recommendation for high-end graphics on the web.

PDF2SVG enables high-quality conversion from PDF to SVG that maintains the original layout of the document and preserves hyperlinks, colors and fonts. The resulting self-contained and compact SVG files can be distributed, viewed, edited, stored, printed, and published onto web sites and CD-ROMs. For quick and easy document navigation and viewing, PDF2SVG can be configured to create page thumbnails as well as an XML summary describing the document components such as metadata, bookmarks, annotations, etc. Using XSLT or any other XML processor, the user can quickly generate custom HTML and JavaScript 'wrappers' that enable the user to browse multi-page documents in any web browser.

Like other PDFTron products, PDF2SVG does not rely on any other third-party software.

PDF2SVG is also available as a software component for embedding into third-party applications. Please see <http://www.pdftron.com/pdf2svg> for more information.

#### 1.1.1 Key Functions

- High quality conversion from PDF (Portable Document Format) to SVG (Scalable Vector Graphics) that maintains the original document layout and preserves hyperlinks, colors and fonts.
  - Font support: Type1, TrueType, Type3 and Type0/CID Fonts
  - Color: ICC, DeviceN, Separation, RGB, CMYK, Indexed, etc.
  - Support for encrypted PDF documents (40 and 128 bit RC4, 128 bit AES).
  - Support for all kinds of patterns, functions, and compression schemes.
  - Support for all annotation types.
- Compression and SVG optimization: Converted documents can be saved as SVGZ compressed files for fast web downloads.
- PDF2SVG allows for generation of stand-alone SVG files (i.e. SVG files with embedded resources) as well as for SVG files with shared resources.
- All text is converted to Unicode dramatically simplifying text manipulation, editing, and searching. PDF2SVG makes every attempt to map text to a Unicode public area, resulting in better repurposing and text-searching capabilities of converted documents.
- Font embedding and font substitution option: The font embedding option allows for reliable and accurate font reproduction. The font substitution option produces smaller files but may not produce 100% accurate fonts on all systems.
- Thumbnail generation.
- Option to extract document-level information that has no counterpart in SVG (e.g. metadata, bookmarks, annotations, etc).
- Batch conversion: Using PDF2SVG you can easily convert single files or whole PDF repositories.
- Efficiency: PDF2SVG is based on PDFNet SDK, making it extremely fast and efficient. The conversion speed is suitable for interactive and dynamic applications.
- Support for all versions of the PDF format (PDF 1.0 to ISO32000).

### 1.1.2 What's New in PDF2SVG?

- Generation of XML summary document describing important document components such as document metadata, bookmarks, annotations, etc. Using XSLT or any other XML processor, XML summary can be used to wrap separate SVG documents into a composite document for e-book style navigation and viewing or for further repurposing.
- Thumbnail generation option for fast navigation through multi-page documents.
- Support for soft, explicit, and color-key masks.
- Better conversion of PDF forms and annotations.
- The conversion speed is significantly improved.
- Smaller file size of generated SVG output.
- SVG output is now faster to render in popular SVG viewers.
- Support for 128-bit AES (Advanced Encryption Standard) encryption and Crypt filters.
- Files with broken cross reference tables are automatically repaired.
- Support for clipping user defined regions.
- Configuration file for frequently used options.

### 1.1.3 Why SVG?

## There are many benefits of converting your documents to SVG:

- SVG is a W3C (Web Standards Consortium) standard format and is backed by a large number of companies and non-profit organizations.
- Free SVG viewers are widely available on major platforms and operating systems.
- Because SVG is based on XML, the document can be easily edited in a text editor. SVG XML content can be linked to back-end business processes such as databases, application servers, and other rich sources of real-time information.
- There are a growing number of affordable and powerful SVG authoring and editing solutions.
- Enhanced search capabilities. All text in SVG is stored in standard XML syntax and Unicode encoding that makes searching operations within a document or across large collections of documents a breeze.

### 1.1.4 Common Use Case Scenarios

- Server-based, on-demand conversion of PDF documents to SVG.
- Batch processing of PDF collections. PDF2SVG is particularly useful in assembling product catalogues and brochures.
- PDF content extraction and repurposing through SVG and XML.

### 1.1.5 Operating Systems Supported

- Windows 7, 2008, Vista, XP, 2003, 2000, NT, 98
- Mac OSX
- Linux

### 1.1.6 System Requirements

- At least 10 MB of free disk space.
- Memory requirement is heavily dependent on the nature of the document being converted into an image file.

## 1.2 About This Manual

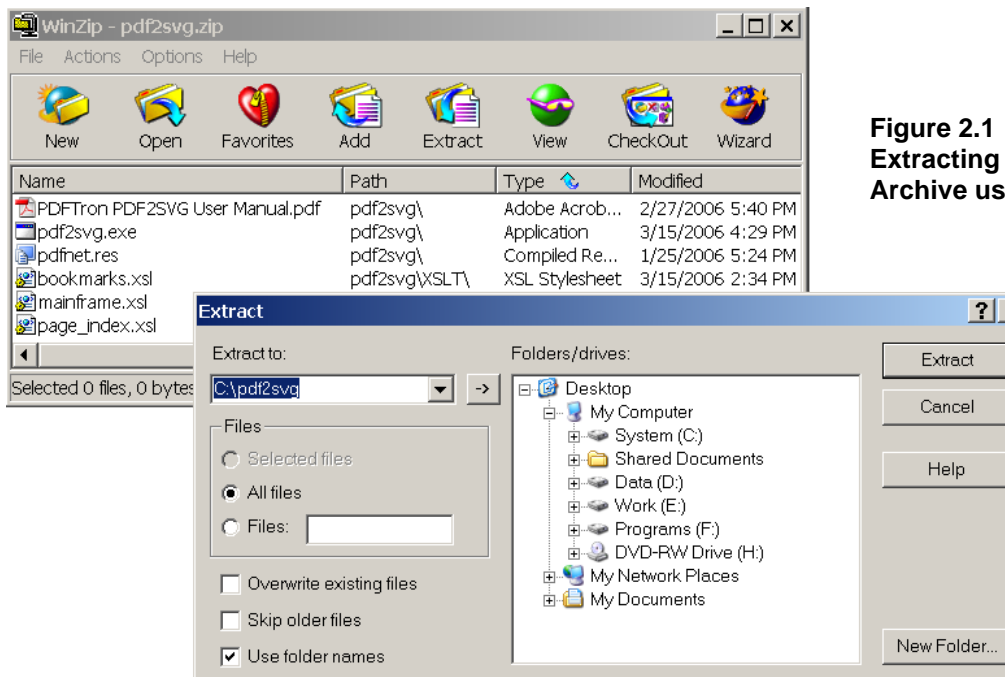
This manual is intended as a guide to the installation and use of PDF2SVG.

- [Section 1](#) introduces PDF2SVG and describes the manual.
- [Section 2](#) explains how to install and uninstall PDF2SVG.
- [Section 3](#) covers basic use of PDF2SVG.
- [Section 4](#) is where you will find all the support information you may require, such as how to report a problem with the software.

## 2. Installing and Uninstalling PDF2SVG

### 2.1 PDF2SVG Installation

PDF2SVG Command-line Application is supplied as a download from a distributor or directly from [www.pdftron.com](http://www.pdftron.com). The release is packaged as a .zip file. (pdf2svg.zip). Unzip the archive in the desired location and make sure to preserve the directory (folder) structure when extracting the archive. In order to register the software, copy the license file provided to you into the “pdf2svg” folder.



**Figure 2.1**  
Extracting PDF2SVG  
Archive using WinZip

### 2.2 Demo Version Installation

If you wish to evaluate the product, you can download the demo version of the product without any serial number or license key.

To do this, go to PDFTron's **Downloads** page at [www.pdftron.com/downloads.html](http://www.pdftron.com/downloads.html). Click on the appropriate product version/name, which will bring you to the product and the appropriate link for the demo download. Simply Download the zip file (pdf2svg.zip) and extract the archive in the desired location, while making sure to preserve the directory (folder) structure when extracting the archive. This will provide you with a working copy of the application. The limitation of the evaluation version is that all pages in processed documents will have a demo stamp.

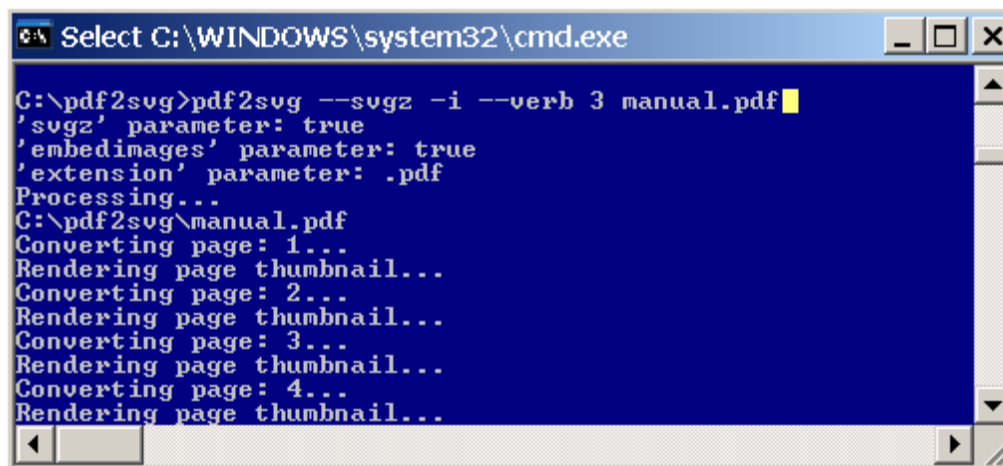
### 2.3 Uninstalling PDF2SVG

To remove PDF2SVG from a computer, simply delete the “pdf2svg” folder.



### 3. Overview

PDFTron's PDF2SVG is a command-line application designed to convert PDF files to SVG, the open-standard W3C recommendation for high-end graphics on the web. The flawless conversion process creates web-ready SVG documents. This section covers the basic use of PDF2SVG explaining all the available options.



**Figure 3.0 PDF2SVG Command-line Application.**

### 3.1 Basic Syntax

The basic command-line syntax is:

```
pdf2svg [options] file1 file2 folder1 file3 ...
```

The following command-line arguments are available for PDF2SVG.

10

		By default, all images are saved as external files.
--nofonts	--nofonts	Disables conversion of font data to SVG. This option will usually result in the smaller SVG file size, but due to font substitution the text may not render accurately.  By default, all available fonts are converted to SVG.
--nounicode	--nounicode	Disables mapping of text to public Unicode region. Instead text will be converted using a custom encoding.  By default, all text is mapped to Unicode.
-b or --box	-b media	Specifies the page box/region to use for clipping. Possible values are: <ul style="list-style-type: none"> <li>■ <b>media</b></li> <li>■ <b>crop</b></li> <li>■ <b>trim</b></li> <li>■ <b>bleed</b></li> <li>■ <b>art</b></li> </ul> The default is page crop region.
-c or --crop	-c 216,522,330,600	User definable crop box to be used as a top level clip region in the output SVG.  By default, the clip region is identical to currently selected page 'box'.
--noclip	--noclip	Disables page clipping. Any content outside of page boundaries will be visible. By default, all pages are clipped using the crop region for the page.
--noannots	--noannots	Disables conversion of form fields and annotations.
--noxmldoc	--noxmldoc	Disables generation of the XML wrapper document.
--thumbsize	--thumbsize 150	The dimension of thumbnail image in pixels.  By default, PDF2SVG will generate 100x100 thumbnails.
--nothumbs	--nothumbs	Disables generation of thumbnail images.
--asv_css_patch	--asv_css_patch	Some versions of the Adobe SVG viewer fail to read valid CSS font properties. As a result, this option was created as a workaround for this bug in the Adobe SVG Viewer. Please note that using this option may cause problems with other SVG viewers.
--noprompt		Disables any user input. By default, the application will ask for a valid password if the password is incorrect.
-p or --pass	e.g. secret or "my pass"	The password for the input file. Not required if the input document is not secured.
--extension	--extension ".pdf"	The default file extension used to process PDF documents. The default is ".pdf".
-h or --help		Print a listing of available options.
-v or --version		Print the version information.
--verb	--verb 2	Set the verbosity level. Valid parameter values are 0, 1, and 2. The higher number results in more feedback. The default is 1.

## 3.3 Basic Usage

### 3.3.1 How to save converted files in a given folder?

By default, PDF2SVG saves converted files in the current working folder. To specify another output location, use the ‘-o’ (or --output) parameter. For example:

```
pdf2svg -o "c:\My Output" 1.pdf 2.pdf 3.pdf
```

Note: If the specified path does not exist, PDF2SVG will attempt to create the necessary folders.

### 3.3.2 How can I control the output names of generated files?

By default, PDF2SVG creates a separate SVG file for every page in the document. The output filename is constructed using the name of the input PDF file, the page number, and appropriate file extension (i.e. svg or svgz). For example, the following command-line generates a sequence of SVG files starting with mydoc\_1.svg, mydoc\_2.svg, etc.:

```
pdf2svg mydoc.pdf
```

PDF2SVG allows output filename customizations using the ‘--prefix’ and ‘--digits’ options. For example, the following command-line generates a sequence of SVG files starting with newname\_0001.svg, newname\_0002.svg, etc.:

```
pdf2svg --prefix newname --digits 4 mydoc.pdf
```

The ‘--digits’ parameter specifies the number of digits used in the page counter portion of the output filename. By default, new digits are added as needed, but the ‘prefix’ parameter could be used to format the page counter field to a uniform width (e.g. myfile0001.svg, myfile0010.svg, instead of myfile\_1.svg, myfile\_10.svg, etc).

To avoid any ambiguities in file naming, the prefix option should be used only for conversion of individual documents.

### 3.3.3 How do I create compressed SVG (SVGZ)?

To create compressed SVG (SVGZ), use ‘--svgz’ as one of the command-line options. This option will instruct PDF2SVG for compress SVG using GZIP compression and to generate output files with the ‘svgz’ extension. For example,

```
pdf2svg --svgz in.pdf
```

### 3.3.4 How do I produce stand-alone SVG?

Some PDF documents use many small bitmaps to represent text or patterns. In this case, the converted SVG document will reference hundreds of external images. You may choose to embed these images within the SVG document using the ‘--embedimages’ or ‘(-i)’ option.

By embedding images it is possible to create self contained SVG files (i.e. files without any references to external resources). Although it is sometimes desirable to create self contained files, this option can result in files that are slower to render in some viewers. The files with embedded

images may also be slower to download over the Net, and because images can't be shared among different pages the total file size for the entire document may increase.

### 3.3.5 How do I open a password protected PDF?

PDF2SVG will, without user intervention, decrypt and convert documents secured with a master/owner password. If the document is secured using a user (or file open) password, PDF2SVG will prompt you to enter the password.

For unattended conversion, the password can also be specified directly on the command-line using the '-p' (or --password) option. For example:

```
PDF2SVG -p secret secured.pdf
```

The above command line will convert PDF to SVG and will use the provided password ('secret') to open the secured document (i.e. 'secured.pdf').

Note: PDF2SVG supports all standard security options available in PDF, including 40 and 128 bit RC4 encryption, Crypt filters, and 128 AES (Advanced Encryption Standard) encryption.

### 3.3.6 Why is PDF2SVG generating page thumbnails and the XML summary document?

By default, PDF2SVG generates a bitmap thumbnail for each converted SVG and one XML summary document for the entire document. Image thumbnails can be used for quick preview of SVG documents, whereas XML summary document could be used to create HTML files that wrap SVG files in a web ready eBook. XML summary document can also be used for content repurposing, navigation, and indexing.

PDF2SVG lets you control the dimensions of thumbnail images using '--thumbsize' parameter. The following command-line will generate 150x150 pixel image thumbnails for every page in the document:

```
pdf2svg --thumbsize 150 in.pdf
```

To disable generation of thumbnail images, use the '--nothumbs' option. Similarly, to disable generation of wrapper XML, use the '--noxmldoc' switch.

### 3.3.7 How do I specify which pages to convert?

By default, PDF2SVG will convert all pages. You can specify a subset of pages to convert using the '-a' or '--pages' options. For example:

```
pdf2svg -a 1,3,10 in.pdf
```

will convert only pages 1, 3, and 10. Please note that PDF2SVG assumes that all pages are numbered sequentially starting from page 1.

To specify a range of pages, use dash character between numbers. For example:

```
pdf2svg -a 1,10-20,50- in.pdf
```

will convert the first page, pages in the range from 10 to 20 and all pages starting with page 50 to the last page in the document.

```
pdf2svg --pages even in.pdf
```

```
pdf2svg --pages odd,100- in.pdf
```

PDF2SVG supports batch conversion of many PDF files in a single pass. To convert all PDF files in a given folder(s) you can use the following syntax:

```
pdf2svg myfolder1
```

```
pdf2svg --subfolders myfolder1 myfolder2
```

```
pdf2svg --extension .blob --subfolders myfolder1
```

A PDF page can define as many as five separate boundaries to control various aspects of the imaging process:

- The media box defines the boundaries of the physical medium on which the page is to be printed. It may include any extended area surrounding the finished page for bleed, printing marks, or other such purposes. It may also include areas close to the edges of the medium that cannot be marked because of physical limitations of the output device. Content falling outside this boundary can safely be discarded without affecting the meaning of the PDF file.
- The crop box defines the region to which the contents of the page are to be clipped (cropped) when displayed or printed. Unlike the other boxes, the crop box has no defined meaning in terms of physical page geometry or intended use; it merely imposes clipping on the page contents. The default value is the page's media box.
- The bleed box defines the region to which the contents of the page should be clipped when output in a production environment. This may include any extra bleed area needed to accommodate the physical limitations of cutting, folding, and trimming equipment. The default value is the page's crop box.
- The trim box defines the intended dimensions of the finished page after trimming. It may be smaller than the media box to allow for production related content, such as printing instructions, cut marks, or color bars. The default value is the page's crop box.
- The art box defines the extent of the page's meaningful content (including potential white space) as intended by the page's creator. The default value is the page's crop box.

```
pdf2svg --box media in.pdf
```





PDF2SVG supports processing of multiple input documents in the same run. For example, it is possible to specify multiple PDF folders and PDF2SVG will automatically process all PDF documents matching a given file extension. For example, the following command-line will process all PDF documents in folders 'test1' and 'test2'

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

```
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
```

```
c:\>pdf2svg -o c:/output_folder c:/test1/*.pdf
```

```
pdf2svg-o c:/output_folder c:/test1/A*.pdf
```

```
pdf2svg -o c:/output_folder c:/test1/*1.pdf
```

The wildcards are expanded in the same manner as operating system commands. (See 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.

To provide additional feedback, PDF2SVG 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.

Exit Code	Description
0	All files converted successfully.
1	Error opening or reading the input file(s).
2	Document is secured. Need a valid password to open the document.
3	There was an error during conversion.
4	An unknown exception encountered.

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



### 3.7.7 Why is a white space separating neighboring pictures?

### 3.7.8 Can I integrate PDF2SVG with my client/server application?

### 3.7.9 Does PDF2SVG have any dependencies on third party components/software?

20

### 3.8 XML Summary Document

This section describes the XML Summary Document that can be generated using PDF2SVG and its potential use in various applications.

By default PDF2SVG generates an XML Summary Document for every PDF document. The XML Summary Document contains document-level information that is not part of SVG files that describe individual pages. The information includes general information about the document (such as author, subject, title, keywords), as well as a listing of document parts and relationships such as pages, thumbnails, annotations, and bookmarks.

The following is a sample XML snippet generated by converting this user manual to SVG:

```
<?xml version="1.0"?>
<!-- Generator: PDFTron PDF2SVG Converter -->
<doc name="1" ext="svg">
  <info>
    <title>PDFTron PDF2SVG User Manual</title>
    <author>PDFTron Systems</author>
    <subject>PDFTron PDF2SVG User Manual</subject>
    <keywords></keywords>
    <creator>Acrobat PDFMaker 7.0.7 for Word</creator>
    <producer>Acrobat Distiller 7.0.5 (Windows)</producer>
  </info>
  <pages>
    <page id="1" href="1_1.svg">
      <thumb href="1_1_thumb.jpg"/>
    </page>
    <page id="2" href="1_2.svg">
      <thumb href="1_2_thumb.jpg"/>
    </page>
    ...
  </pages>
  <bookmarks>
    <bookmark title=" 2. Installing and Uninstalling PDF2SVG" open="true"
goto="7" href="1_7.svg">
      <bookmark title="2.1 PDF2SVG Installation" open="false" goto="7"
href="1_7.svg"/>
      <bookmark title="2.2 Demo Version Installation" open="false" goto="7"
href="1_7.svg"/>
      <bookmark title="2.3 Uninstalling PDF2SVG" open="false" goto="7"
href="1_7.svg"/>
    </bookmark>
    ...
  </bookmarks>
</doc>
```

Most of the elements and attributes are self explanatory. The 'info' element lists document information properties, the 'pages' element lists all 'page' elements that are part of the high level 'document', and the 'bookmarks' element specifies the outline tree that can be used for quick navigation between pages.

The summary document can be used as a map of the abstract document that contains many SVG files representing document pages, as well as outline tree and annotations describing how different document parts are related.

In most cases, the summary document is further consumed by an XML consumer/processor (e.g. XML DOM/SAX Library or XSLT). For example, an application may read XML summary to create database records for archiving purposes. Another application may implement interactive navigation through SVG pages using the document outline.

Yet another example of the XML wrapper consumer is an eBook generator that converts the XML Summary Document to HTML. The generated HTML would wrap converted SVG files and would provide web-based eBook interface for navigation between different pages, including bookmark tree, thumbnail index, etc. The end result would look like what is illustrated in the following figure:

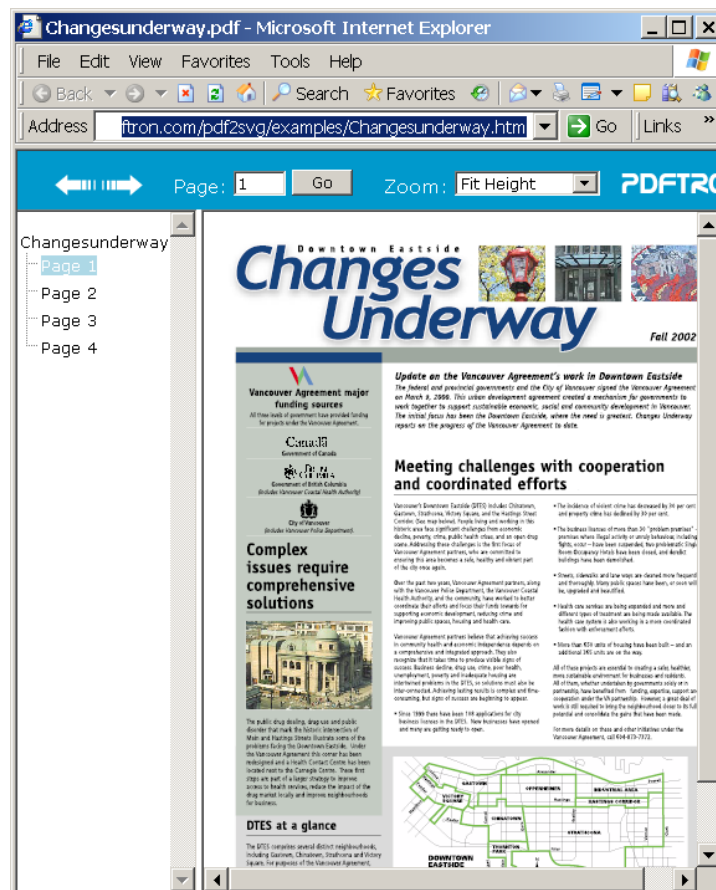
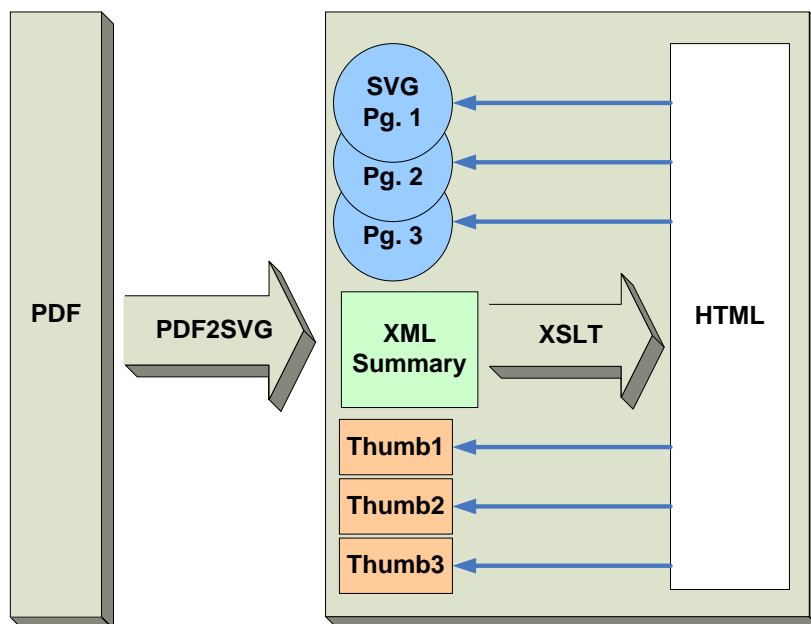


Figure: SVG wrapped in an HTML web-browser eBook.

The process used to create HTML eBook wrapping converted SVG-s is illustrated in the following figure:



Using PDF2SVG, a PDF document is converted to a set of SVG images and their thumbnails, as well as the XML Summary Document. The fastest way to create HTML wrappers around SVG is using XSLT. XSLT is a very simple language for transforming XML documents. A simple XSLT transform may look as follows:

```
<?xml version='1.0'?>
<xsl:stylesheet version='1.0'
xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
  <xsl:output method='html' indent='yes' doctype-public='-//W3C//DTD HTML
3.2 Final//EN' />
  <xsl:template match='/'>
    <HTML>
      <HEAD>
        <TITLE>HTML SVG Wrapper</TITLE>
      </HEAD>
      <BODY>
        <xsl:apply-templates select='doc/info' />
        <HR />
        <xsl:apply-templates select='doc/pages' />
      </BODY>
    </HTML>
  </xsl:template>

  <xsl:template match='info'>
    <table border="0" cellspacing="0" cellpadding="4">
      <tr><td>Title:</td><td><xsl:value-of select='title' /></td></tr>
      <tr><td>Author:</td><td><xsl:value-of select='author' /></td></tr>
      <tr><td>Subject:</td><td><xsl:value-of select='subject' /></td></tr>
      <tr><td>Keywords:</td><td><xsl:value-of select='keywords' /></td></tr>
      <tr><td>Creator:</td><td><xsl:value-of select='creator' /></td></tr>
      <tr><td>Producer:</td><td><xsl:value-of select='producer' /></td></tr>
    </table>
```

```

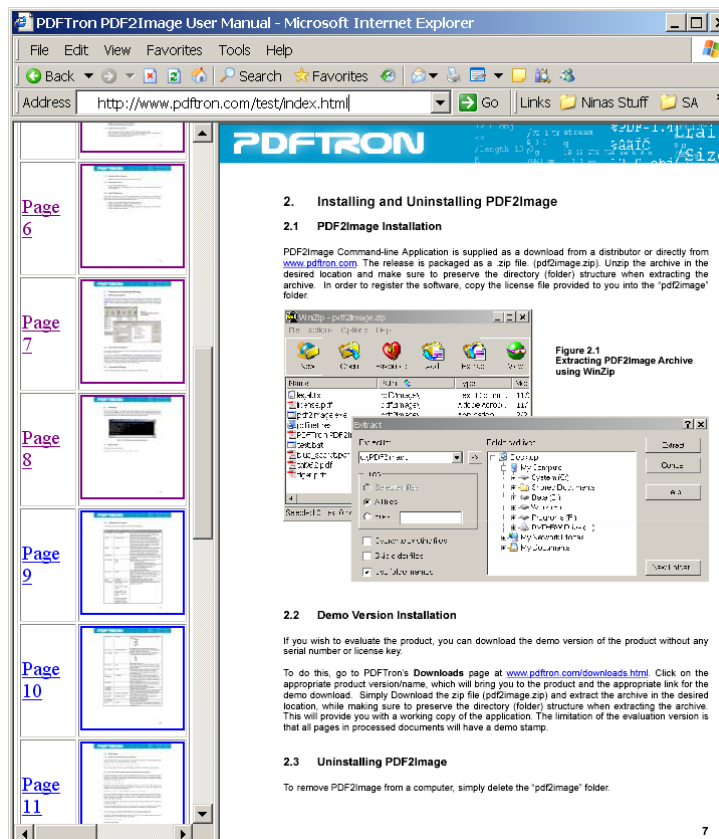
</xsl:template>

<xsl:template match='pages'>
  <TABLE BORDER="1">
    <xsl:apply-templates/>
  </TABLE>
</xsl:template>

<xsl:template match='page'>
  <TR>
    <TD><A TARGET="view" HREF="{@href}">Page <xsl:value-of
select='@id' /></A></TD>
    <TD><A TARGET="view" HREF="{@href}"><IMG
SRC="{thumb/@href}" /></A></TD>
  </TR>
</xsl:template>
</xsl:stylesheet>

```

The above XSLT template will create an HTML page containing general information about the documents such as it title, subject, keywords, etc. The HTML will also contain a thumbnail index of all pages in the document. Clicking on page labels or on thumbnails will open SVG graphics in the right pane of the browser window. The final result would look as follows:



To run XSLT transforms you can use your favorite XSLT processor. As a starting point, PDF2SVG distribution comes with a sample project illustrating how to run XSLT transform using Microsoft .NET Framework.



## 4. Support

### 4.1 Reporting Problems

If you encounter a problem or question regarding PDFTron PDF2SVG which is not addressed in this manual or on PDFTron's website, please submit a problem report to PDFTron's Support Group at <http://www.pdftron.com/reportproblem.html>.

When submitting a problem you will be asked to provide the following information:

- Contact details
- Product and Version of the product
- Detailed description of problem
- Problem file(s)
- Whether you have an AMS (Annual Maintenance Subscription)
- Any other information that may be related

### 4.2 Contact Information

To contact PDFTron directly, please use the contact information below:

Tel: 1-604-730-8989  
Fax: 1-604-676-2477

Web site: [www.pdftron.com](http://www.pdftron.com)

Email Contacts:

General Business Inquiries: [info@pdftron.com](mailto:info@pdftron.com)  
Sales & Licensing: [sales@pdftron.com](mailto:sales@pdftron.com)  
Product Support: [support@pdftron.com](mailto:support@pdftron.com)  
Professional Services: [services@pdftron.com](mailto:services@pdftron.com)  
Website related questions: [webmaster@pdftron.com](mailto:webmaster@pdftron.com)  
Press & News: [press@pdftron.com](mailto:press@pdftron.com)