

wsW2LTX 1.1 API

Ingo H. de Boer

May 25, 2006

email: <mailto:idb@winshell.org>, url: <http://www.winshell.org/>

Copyright © 2005-2006 Ingo H. de Boer.

Permission to use, copy and distribute this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice, author statement and this permission notice appear in all copies of this software and related documentation.

THE SOFTWARE IS PROVIDED 'AS-IS' AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL INGO H. DE BOER OR ANY PERSON OR INSTITUTION RELATED TO INGO H. DE BOER BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Contents

1	Introduction	1
2	Library and header files	1
3	API reference	2
3.1	Required methods	2
3.1.1	SetInputName(char* pchInputName)	2
3.1.2	Convert()	2
3.2	Optional methods	2
3.2.1	ErrorOut(char* pchErrorMessage)	2
3.2.2	SetOutputName(char* pchOutputName)	2
3.2.3	SetOutputDirectory(char* pchOutputDirectory)	3
3.2.4	SetGSFontDirectory(char* pchGSFontDirectory)	3
3.2.5	SetGSFontMapName(char* pchGSFontMapName)	3
3.2.6	SetCharSet(char* pchCharSet)	3
3.2.7	SetPassword(char* pchPassword)	3
3.2.8	NoGraphics(bool bNoGraphics)	3
3.3	Static methods	3
3.3.1	GetVersion(char *pchProductVersion, char *pchFileVersion)	3
4	Example of using the wsW2LTX library	4

1 Introduction

The wsW2LTX library is an application programming interface (API) designed to translate a MS Word document to \LaTeX .

The library is based on libwv and on several other – mostly Unix based – libraries. These are in detail:

libwv	http://wvware.sourceforge.net/
libwmf	http://wvware.sourceforge.net/
freetype	http://freetype.sourceforge.net/
gdk-pixbuf	http://developer.gnome.org/arch/imaging/gdkpixbuf.html
glib	http://www.gtk.org/
gd	http://www.boutell.com/gd/
iconv	http://www.gnu.org/software/libiconv/
gettext	http://www.gnu.org/software/gettext/
imagemagick	http://www.imagemagick.org/
libpng	http://www.libpng.org/
zlib	http://www.zlib.net/

My thanks go to the developers and supporters of the libraries mentioned above.

2 Library and header files

The package consists of three libraries and a header file. These are in detail:

1. Header

- **wsW2LTX.h**: Include this file into your project.

2. Library

Decide which compilation model you will use and link against one of the following libraries:

- **wsW2LTXML.lib**: Single-threaded, statically linked, ML run-time
- **wsW2LTXMT.lib**: Multi-threaded, statically linked, MT run-time
- **wsW2LTX.lib**: Multi-threaded, dynamically linked, MD run-time
Here, ship your executable file with **wsW2LTX.dll**.

For a statically linked project, it must be compiled with the `WS_W2LTX_STATIC` flag.

3 API reference

The `wsW2LTX` library exports a class called `'wsW2LTX'`. This class has required, optional and static methods which are described in the following. A detailed C++ example can be found in the section 4.

3.1 Required methods

3.1.1 `SetInputName(char* pchInputName)`

This method sets the input file name, the MS Word document.

3.1.2 `Convert()`

This method does the actual conversion from the MS Word document to a `LATEX` document. The input file name has to be set first.

3.2 Optional methods

3.2.1 `ErrorOut(char* pchErrorMessage)`

This method is virtual and can be overwritten from a class based on `wsW2LTX` to redirect the output to a window or a file. An example is:

```
void my_wsW2LTX::ErrorOut(char* pchErrorMessage)
{
    ASSERT(m_pDialog);
    CString cstrError = pchErrorMessage;
    cstrError = _T(" info: ")
                + cstrError.Left(cstrError.GetLength() - 1);
    m_pDialog->AddLog(cstrError);
}
```

By default, the error information is written to `stderr`.

3.2.2 `SetOutputName(char* pchOutputName)`

Normally, the output name is created from the input name by changing the extension from `.doc` to `.tex`. This behaviour can be changed by setting the output file name with this method.

3.2.3 SetOutputDirectory(char* pchOutputDirectory)

Normally, the output file(s) are written in the same directory where the MS Word document resides. With this method, the output files are written to the directory given by `pchOutputDirectory`.

3.2.4 SetGSFontDirectory(char* pchGSFontDirectory)

The `wsW2LTX` package comes with the `AFPL Ghostscript Fonts` which are stored in the directory `gs_fonts`. This directory can be changed with this method.

3.2.5 SetGSFontMapName(char* pchGSFontMapName)

The `AFPL Ghostscript Fonts` fontmap name is `fonts.dir`. The name of this file can be changed with this method.

3.2.6 SetCharSet(char* pchCharSet)

The library tries to find out the character set of the MS Word document automatically. If this fails, the character set can be set with this method. The name must correspond to one of the names from the `iconv` library, e. g. `"UTF-8"` or `"ISO-8859-15"`.
<http://www.gnu.org/software/libiconv/>

3.2.7 SetPassword(char* pchPassword)

If the MS Word document is password protected the password can be set with this method.

3.2.8 NoGraphics(bool bNoGraphics)

This method tells the library to produce no images at all.

3.3 Static methods

3.3.1 GetVersion(char *pchProductVersion, char *pchFileVersion)

This static method gives information about the version of the library, e. g.:

```
std::string strFileVersion;  
std::string strProductVersion;  
wsW2LTX::GetVersion(&strProductVersion[0], &strFileVersion[0]);  
cout << "wsW2LTX version: " << &strFileVersion[0] << endl;
```

4 Example of using the wsW2LTX library

The following lines describe the usage of the library in C++. Only three steps will convert a MS Word document to a \LaTeX document:

1. Create the wsW2LTX object.
2. Set the MS Word file name.
3. Call `Convert()`.

```
01: #include "wsW2LTX.h"
02:
03: void usage(char *name)
04: {
05:     cout << "Usage: " << name << " filename.doc" << endl;
06:     exit (-1);
07: }
08:
09: int _tmain(int argc, _TCHAR* argv[])
10: {
11:     if (argc < 2)
12:         usage (argv[0]);
13:
14:     // create wsW2LTX object
15:     wsW2LTX _wsW2LTX;
16:
17:     // set input name of file to convert
18:     _wsW2LTX.SetInputName(argv[1]);
19:
20:     // convert the file
21:     cout << "converting file: " << argv[1] << endl;
22:     if (!_wsW2LTX.Convert())
23:         cerr << "could not convert the document!" << endl;
24:     else
25:         cout << "finished converting..." << endl;
26:
27:     return 0;
28: }
```

This sample is also included into the wsW2LTX library package.