

Legimi S.A.

Legimi Sp. z o.o.

Legimi Watermark

Technical documentation.

February 22, 2018

Contents

1	Introduction	3
1.1	Purpose of this document	3
1.2	Product scope	3
1.3	Document description	3
2	Product description	3
2.1	Functionality	4
2.2	Functioning	4
2.3	Durability	4
2.4	Correctness and compatibility with readers' software	4
2.5	Licensing	5
2.6	Deployment	6
2.7	Manual	6
2.8	Errors	8
2.9	Limitations	9
A	Appendix	10
A.1	Usage examples	10
A.2	Compiling Mono for Red Hat	10

1 Introduction

1.1 Purpose of this document

This document presents functionality and usage of console tool for electronic publication securing with watermark (EPUB, MOBI, PDF).

1.2 Product scope

Product described in this document deals with a problem of intellectual property protection through placing an electronic mark in publication. Unlike the DRM system, user does not need a special program or a specific set of devices to read publications protected with watermark. Instead of encrypting data, publication receives additional information which let its owner to be identified.

Legimi Watermark allows applying watermark on the file and detecting a watermark already applied in the data. Depending on the file format there are different methods used for applying a piece of information about an owner. Watermark is created both in visible (short message about data protection) and invisible way for users.

1.3 Document description

Further in this document we will describe functionality, set-up and interpretation of information signs. We will also indicate technical limitations. Documentation concerns *Legimi Watermark* tool in version 3.0.0 or newer.

2 Product description

Legimi Watermark is a console tool prepared for implementation on content servers. It works in two modes:

- encoding mode: on the basis of EPUB, MOBI or PDF file and user information program generates other file in similar format to input file. Output file includes encoded information about purchaser.
- decoding mode: program reads watermark information from protected EPUB or MOBI files. If there is no watermark, program informs about it.

Legimi Watermark supports following input file formats:

- EPUB version 2
- EPUB version 3
- MOBI version 6
- hybrid MOBI files, containing version 6 and version 8 data (those are the file created by default by *Kindlegen*)
- PDF version 1.7 or older (ISO 32000-1)

2.1 Functionality

Watermark technology should be used while downloading purchased publication by the buyer. Supposing, that original file and transaction number remain unchanged in two different sessions, the watermarked file will be the same. It gives an opportunity to cache watermarked files for each transaction, but this functionality is not a part of *Legimi Watermark* solution and should be developed separately.

2.2 Functioning

Way of applying watermark depends on input file. For EPUB and MOBI files information is coded through several different and independent channels. Thanks to this, deleting of watermark by an unauthorized person is complicated. Encoding does not influence visual side of an e-book – watermark is not visible in EPUB and MOBI browsers. To cause an additional psychological effect, there is a visible text at the beginning of publication informing about copyright protection. In case of PDF files, there are visible marks at each page, additional metadata information, and invisible text hidden in page content. PDF support includes also some additional options which allow configuration of visible text appearance.

2.3 Durability

Key feature of watermark technology is the durability of this protection (file is resistant to attempts of removing, in particular with popular applications). *Legimi Watermark* was for this cause tested with *Calibre* (<http://calibe-ebook.com>) version 3.17 and *Kindlegen* (<https://www.amazon.com/gp/feature.html?docId=1000765211>). In both cases tests were performed on Windows 8.1 64 bit workstation.

Calibre is one of most popular programs to manage electronic publications library. Its popularity comes from several files supported and decrypting functionality (after installing a additional plug-in it enables easy removal of Adobe Digital Edition's protection). Tests proved that *Legimi Watermark* is resistant to conversion between MOBI and EPUB in *Calibre*. It means that the *Legimi Watermark* can be still read after the original file was converted from EPUB to MOBI. And watermark in secured files remains invisible after conversion.

Kindlegen is a tool created by Amazon and used primarily by publishers. It allows to create MOBI file by conversion from EPUB input file (there is no EPUB to MOBI conversion). Watermark applied to MOBI file with *Legimi Watermark* software is resistant to conversions made in *Kindlegen*.

At the moment of writing this document there was no publicly available tool, which was able to automatically detect and remove *Legimi Watermark* from protected publications. As all such protections, *Legimi Watermark* may be deleted by manual file edition, but this process involves professional knowledge.

2.4 Correctness and compatibility with readers' software

Legimi Watermark generates correct (that is: consistent with file's format specification) output files, provided input file was also correct. For EPUB and PDF formats there are publicly accessible specification sheets, so output files can be verified using widely available validation software.

MOBI format however, is a closed format owned by Amazon, with no specification officially available. Validation can only be performed indirectly, by using Amazon's *send to kindle* feature (<https://www.amazon.com/gp/sendtokindle>). This Amazon Cloud function allows customer

to attach MOBI file to an email. After email is sent to proper address, file is verified on Amazon's servers; if validation fails, email is received (with no validation details). Otherwise, file will reach its destination device (either Kindle e-reader or smartphone or tablet with Kindle application installed) and file is assumed to be correct.

Files protected with *Legimi Watermark* can be viewed properly on all most popular reading software and e-reader devices. We have positively verified the following applications:

EPUB format readers:

- Adobe Digital Editions 3.0
- Calibre 3.17
- Moon+ Reader 3.5.4
- Google Play Books 4.0.47.184185130
- iBooks dla iOS 11.2.5

MOBI format readers:

- Kindle for PC 1.21.0
- Kindle for Android 7.0.0.49
- Kindle for iOS 6.4
- Kindle 4 e-reader
- Kindle Paperwhite 3 e-reader
- Kindle 8 e-reader

It is highly probable, that *Legimi Watermark* properties are preserved in all modern e-reader platforms. Due to limited resources, it was not possible to test all of hardware and software configurations, though.

2.5 Licensing

There 3 licensing models available for *Legimi Watermark*:

- single payment
- commission payments - commission is paid for each transaction, which results in download of the protected file (*pay per file*)
- commission payments with Software as a Service deployment (*pay per file too*)

In first two cases, deployment procedure is very similar: you need to run *Legimi Watermark* software with proper console line arguments. In case of *pay per file* license, you need to provide some additional parameters, which will be used to calculate commission rate. Running program in this mode, requires connectivity with Legimi servers.

The last option is to add file protection using Software as a Service architecture. Client is able to request file protection (or watermark detection) by a call to the remote service. Legimi is responsible for maintaining this service. Service exposes REST API endpoint. Detailed documentation of the REST API is available as a separate document.

2.6 Deployment

Legimi Watermark does not require any sophisticated configuration - all you need to do is to place program files in chosen directory (a.k.a x-copy deployment). To run the program, you need .NET 3.5 runtime or Mono 2.8 (or newer) to be installed on your machine. On the Windows operating system, .NET environment is installed by default. For computers running other operating systems, Mono runtime installation is required. You can download Mono from <http://www.mono-project.com/download/stable/#download-lin>. Binary distributions are available, among others, for Suse, Ubuntu, Mac OSX and Solaris. In some cases you will need to build Mono from sources. An guide for Red Hat system can be found in appendix.

If you have chosen *pay per file* license, *Legimi Watermark* will connect to Legimi servers in order to send commision information. To allow this, TCP connectivity muyst be enabled to the www.legimi.com server, on port 80. Additionally, program will store some data in its current directory - write priviliges are required.

2.7 Manual

Legimi Watermark program in ran as a console tool. Required arguments (in brackets abbreviation if exist):

- `--mode (-m)`: sets mode. Possible values are *encode (e)* or *decode (d)*.
 - *encode* sets the program in encoding mode which means applying watermark to the electronic publication. In this mode the result is a file saved in indicated localization.
 - *decode* sets the program in decoding mode which means reading a watermark text from a electronic publication. If a watermark is found, it will be written to a standard output.
- `--type (-t)`:sets the input file format. Possible settings are *epub*, *mobi* or *pdf*.
- `--input (-i)`: path to the input file
- `--password (-p)`: password used to encrypt (or decrypt) the watermark. The use of a password prevents from reading of the watermark information by third parties, even if they use Legimi (it is possible only to determine the presence of the watermark). Password should consist of up to 32 ASCII characters. For PDF files this applies only if you intend to limit user's right to copy or print the document. Otherwise password will be ignored for PDF files.

Additional arguments required in coding mode (they are ignored in decoding mode):

- `--output (-o)`: output path
- `--message (-msg)`: watermark text which is encoded in output file. Watermark should consist of max. 48 ASCII marks. Exception to this rule are PDF files which are not limited in for watermark length.
- `--no-transparent`: one of watermarking technology for EPUB and MOBI files is adding of transparent texts at the end of chapters (right before tag `mbp:pagebreak` in case of MOBI file). Some browsers (FBReader, Kindle for PC) work with transparent texts improperly and hidden message is visible. This option lets switch off this way of applying a watermark. Information about protection stays invisible, but the protection itself becomes weaker. This option has no effect on PDF files.

Depending on input file format there are also possible additional arguments for particular file formats. For EPUB files:

- *--epub-watermark-page*: path to XHTML file, which is added to EPUB file as an information page about watermark protection. This XHTML is placed on the second page of EPUB file, after the cover. By default the *default-info.xhtml* file is used. This file is a part of Legimi application. You can use any valid XHTML file. Any external references (like images or stylesheets) will automatically added to the EPUB file.
- *--epub3-watermark-page*: same as above option, however the file given is only used if the input file is in the EPUB 3 format.
- *--epub-no-watermark-page*: : choosing of this option causes no information page (about watermark protection)
- *--epub-encoding-heuristic*: by default, if html files of EPUB file do not declare a charset, some of watermark coding may not be used. This option uses heuristic methods to determine file charset.
- *--epub-encoding-force*: forces give file encoding, if HTML file has no charset defined. This is an alternative to *--epub-encoding-heuristic*. If both options are supplied, heuristic will not be used.

For PDF files:

- *--pdf-font*: a given font will be used in a information message (about watermark protection) in PDF-file. Possible settings: courier, courier-bold, Helvetica, Helvetica-bold, times, times-bold, times-italic, times-italic-bold. By default Legimi Watermark uses Helvetica font.
- *--pdf-font-size*: font size using in watermark message for PDF file. By default 16.
- *--pdf-font-color-r*: red font, in RGB system. By default 0 (black).
- *--pdf-font-color-g*: green. By default 0 (black).
- *--pdf-font-color-b*: blue. By default 0 (black).
- *--pdf-no-copy*: disables user ability to copy content from the PDF file. To use this option, you also need to provide password with *--password* parameter. By default copying is allowed.
- *--pdf-no-print*: disables user ability to print the PDF file. To use this option, you also need to provide password with *--password* parameter. By default printing is allowed.
- *--pdf-text-position*: sets the distance from the bottom of page to the first line of text, in pixels. Fractional numbers are allowed, with dot as a decimal separator. Only non negative numbers are valid.

For MOBI files:

- *--mobi-watermark-message*: is for specify an information text about watermark protection. It will be placed at the beginning of an output file and will be displayed by readers which use version 6 of MOBI file. Text should be a proper html tag. It is possible to use CSS-style, but one should remember that MOBI format does not support html and css standards in full. By default the following text is used: `<p>Plik jest zabezpieczony znakiem wodnym</p>`.

- *--mobi-no-watermark-message*: information message will not be added to MOBI-file.
- *--mobi-kf8-watermark-message*: is for specify an information text about watermark protection. It will be placed at the beginning of an output file. Text should be a proper html tag. It will be displayed by readers which use version 8 of the MOBI file (Kindle Fire 8).
- *--mobi-leave-srcs*: Many MOBI files contain SRCS record, where original EPUB file can be found. This is the standard behavior of *Kindlegen*, so most MOBI files generated by this application will have SRCS record. *Legimi Watermark* removes this record by default - this greatly reduces output file's size and prevent leak of unprotected EPUB file from SRCS record. You can use *mobi-leave-srcs* option to disable this behavior and leave SRCS record intact.

Sample commands to run this program were attached to this document in appendix.

2.7.1 Pay per file licensing

In case of deployment in pay per file mode, there some additional parameters, which must be supplied to the command line, for any type of input file:

- *-transaction*: unique identifier of the transaction, for which the file is generated
- *-price*: gross value of the book being protected. Use dot as a decimal separator.

Above information is emmediately send to Legimi service. To optimize performance, communication with Legimi server is peformed in parallel with protection process. In case of connectivity errors, output file will not be saved and program will exit with error.

2.8 Errors

Program informs about errors by showing messages on standard error output, one error in one output line. Type of error can be identified by the beginning of the message:

- *WARN* - indicates a warning, which probably will not prevent continued operation of the program. This may be providing of too many arguments while running the program (i.e. providing an output file in decoding mode) or detect minor errors in the input file.
- *ERROR* - indicates an error, which causes break in execution. It may be due to the lack of required arguments or invalid input file.
- *FATAL* - indicates an critical error, which terminates the program. This kind of errors should be reported to Legimi support.

Information about an error may be also checked by inspecting program exit code:

- 0: operation succeeded
- 1: invalid command line arguments
- 2: error
- 3: critical error

2.9 Limitations

- small MOBI and EPUB files: because of the need to make the watermark invisible to the end user, the message should be properly distributed in the whole file. This may be impossible in a small text without pictures.
- compatibility with ereaders: although every effort has been made to verify protected files on wide spectrum of devices and readers' software, there is no way to guarantee proper presentation in every situation. Possible reasons for invalid publication display include malformed fonts, bug in reader's application or uncommon user preferences.

A Appendix

A.1 Usage examples

Protecting an EPUB file:

```
watermark.exe -t epub -m encode -msg 'my message' -i in.epub -o out.epub
```

Reading watermark from an EPUB file:

```
watermark.exe -t epub -m decode -i in.epub
```

Protecting an EPUB file and using a password:

```
watermark.exe -t epub -m encode -msg 'my message' -i in.epub -o out.epub -p secret
```

Reading watermark from an EPUB file, with a password:

```
watermark.exe -t epub -m decode -i in.epub -p secret
```

Protecting a MOBI file, with non default information text:

```
watermark.exe -t mobi -m encode -msg 'my message' -i in.mobi -o out.mobi --mobi-watermark-message '<p>some text</p>'
```

A.2 Compiling Mono for Red Hat

```
yum install gcc libtool bison pkg-config libglib2.0-dev gettext make bzip2 g++
```

```
wget http://origin-download.mono-project.com/sources/mono/mono-2.10.2.tar.bz2
```

```
tar xvjf mono-2.10.2.tar.bz2
```

```
cd mono-2.10.2
```

```
./configure --prefix=/opt/mono-2.10
```

```
# this might take 30-60 minutes
```

```
make
```

```
make install
```