

# MarkdownDoc

User Guide

Version: 1.3.6

Author: Tommy Svensson

Copyright © 2012 Natusoft AB

## Table of Contents

<b>1 MarkdownDoc User Guide</b>	<b>1</b>
1.1 Introduction	1
1.1.1 Binaries	1
1.1.2 Thanks	1
1.1.3 Version history	1
1.1.3.1 1.3.6	1
1.1.3.2 1.3.5	2
1.1.3.3 1.3.4	2
1.1.3.4 1.3.3	2
1.1.3.5 1.3.2	2
1.1.3.6 1.3.1	2
1.1.3.7 1.3	2
1.1.3.8 1.2.10	2
1.1.3.9 1.2.9	2
1.1.3.10 1.2.8	3
1.1.3.11 1.2.7	3
1.1.3.12 1.2.6	3
1.1.3.13 1.2.5	3
1.1.3.14 1.2.4	3
1.1.3.15 1.2.3	3
1.1.3.16 1.2.2	3
1.1.4 How markdown is MarkdownDoc ?	3
1.2 File specifications	3
<b>2 Command Line</b>	<b>5</b>
2.1 General	5
2.2 PDF Generator	5
2.3 HTML Generator	6
2.4 Markdown Generator	6
<b>3 Maven Plugin</b>	<b>7</b>
3.1 generatorOptions	7
3.2 pdfGeneratorOptions	7
3.3 htmlGeneratorOptions	10
3.4 mdGeneratorOptions	11
<b>4 Library</b>	<b>13</b>
4.1 Usage	13
4.1.1 Parsers	13
4.1.1.1 se.natusoft.doc.markdown.parser.MarkdownParser	13
4.1.1.2 se.natusoft.doc.markdown.parser.JavadocParser	13
4.1.1.3 se.natusoft.doc.markdown.parser.ParserProvider	14
4.1.2 Generators	14
4.1.2.1 se.natusoft.doc.markdown.generator.PDFGenerator	15
4.1.2.2 se.natusoft.doc.markdown.generator.HTMLGenerator	15
4.1.2.3 se.natusoft.doc.markdown.generator.MarkdownGenerator	15
4.1.2.4 se.natusoft.doc.markdown.util.MDDocFileHandler	15
<b>5 MarkdownDoc Editor</b>	<b>16</b>
5.1 Features	16
5.1.1 Styling as you type	16
5.1.2 HTML Preview	16
5.1.3 Editing effects	16
5.1.4 Generate PDF & HTML	16
5.1.5 Configurable	17

5.1.6 Load file by drag & drop	17
5.1.7 Special preview drag & drop feature	17
5.1.8 Mac OS X Fullscreen support	17
5.2 Running	17
5.3 Requirements	17
5.4 Functions	17
5.5 If you're on a Mac	18
5.6 Currently Missing	18
5.7 Bugs	18
5.7.1 By me	18
5.7.2 By Oracle	18
<b>6 The mddoc file type</b>	19
6.1 .mddoc format (myfile.mddoc)	19
<b>7 Simple Markdown Reference</b>	22
7.1 Headings	22
7.2 Paragraphs	22
7.3 Italics	22
7.4 Bold	22
7.5 Blockquote	22
7.6 Lists	22
7.6.1 Unordered lists (* or -)	22
7.6.2 Ordered list (n.)	23
7.7 Code block (pre formatted with a fixed width font)	23
7.8 Horizontal rule	23
7.9 Links	23
7.10 Images	23
7.11 backslash (\)	24
<b>8 Licenses</b>	25
8.1 Project License	25
8.2 Third Party Licenses	25
<b>9 License Texts</b>	26
9.1 Apache License version 2.0, January 2004	26
9.1.1 APPENDIX: How to apply the Apache License to your work.	28
9.2 GNU Affero General Public License version v3	28
9.2.1 GNU AFFERO GENERAL PUBLIC LICENSE	28
9.2.2 TERMS AND CONDITIONS	29
9.2.2.1 0. Definitions.	29
9.2.2.2 1. Source Code.	29
9.2.2.3 2. Basic Permissions.	30
9.2.2.4 3. Protecting Users' Legal Rights From Anti-Circumvention Law.	30
9.2.2.5 4. Conveying Verbatim Copies.	31
9.2.2.6 5. Conveying Modified Source Versions.	31
9.2.2.7 6. Conveying Non-Source Forms.	31
9.2.2.8 7. Additional Terms.	32
9.2.2.9 8. Termination.	33
9.2.2.10 9. Acceptance Not Required for Having Copies.	33
9.2.2.11 10. Automatic Licensing of Downstream Recipients.	34
9.2.2.12 11. Patents.	34
9.2.2.13 12. No Surrender of Others' Freedom.	35
9.2.2.14 13. Remote Network Interaction; Use with the GNU General Public License.	35
9.2.2.15 14. Revised Versions of this License.	35
9.2.2.16 15. Disclaimer of Warranty.	36
9.2.2.17 16. Limitation of Liability.	36

9.2.2.18 17. Interpretation of Sections 15 and 16.	36
9.2.3 How to Apply These Terms to Your New Programs	36

# 1 MarkdownDoc User Guide

## 1.1 Introduction

---

MarkdownDoc is a tool that basically does what the name sounds like. My intention with this tool was to be able to document my java opensource tools in markdown and be able to generate both html and PDF from it using a maven plugin.

So why not use mavens site plugin which does support markdown ? These days generating a whole site for your project seems a bit much. Both Bitbucket and GitHub supports markdown documentation right off in a nice and easy way. I want to choose where to put my documentation (ok, most locations in maven can be configured) and I also had the following requirements:

- Be able to generate one PDF document from a whole collection of separate markdown documents so that I can spread them out in different subproject for multi maven project projects. If you are reading this in PDF format this PDF have been put together from multiple sources.
- Be able to generate a table of contents and a title page.
- I just wanted to do it my way OK! :-)

It does also provide a java -jar executable variant. The main functionality is available as a library.

In short MarkdownDoc provides the following:

- Markdown document model.
- Markdown parser.
- Javadoc comment parser.
- PDF generator.
- HTML generator.
- Markdown generator.
- java -jar commandline executable.
- Markdown editor that formats Markdown while writing with preview and PDF + HTML generation. Can be run with java -jar.
- Maven plugin.

### 1.1.1 Binaries

Version 1.2.9 is available in maven central.

Binaries for newer versions will be made available at [Bintray](http://jcenter.bintray.com) and Bintrays JCenter repository: <http://jcenter.bintray.com> which should also contain everything in maven central.

### 1.1.2 Thanks

Thanks to John Gruber for the brilliant markdown document format, and to iText Software Corp. for making an excellent easy to use PDF library and making it available as open source.

### 1.1.3 Version history

#### 1.1.3.1 1.3.6

Bug fixes in MarkdownDocEditor:

- Preformatted styling should now behave correctly.
- Preformatted font (monospace) settings now work. Also defaulted font size of monospace to 14 rather than 16.

### 1.1.3.2 **1.3.5**

What I did not mention in the information for version 1.3.4 is that the editor was converted from Java to Groovy. Here I apparently ran into a Groovy gotcha: What looked to be a member reference were actually a property reference to the same method that tried to reference member. In this case it was an anonymously implemented interface with a getter whose implementation tried to reference the outer class member of same name as getter property, and got the property rather than the member causing a never ending loop resulting in `java.lang.StackOverflowError`.

This affected only generating of PDF and HTML. The error occurred after writing generated output, but before opening the generated output (when told to do so by checkbox setting). This problem is now fixed by this version and is the only thing that differs from previous version.

### 1.1.3.3 **1.3.4**

Fixed a bug with relative path for images using *PDFGenerator* reported by Maher Gamal. There are now 5 ways to specify paths to images for PDF:

1. Absolute path
2. Relative to current directory.
3. Relative to markdown document.
4. Relative to resulting PDF document.
5. Relative to a supplied root dir. This can now be specified in the PDF generator options. If using the library, passing `rootDir` will override the options `rootDir`.

These paths will be automatically resolved.

### 1.1.3.4 **1.3.3**

Ironed out all *known* bugs in editor.

### 1.1.3.5 **1.3.2**

Added markdown formatting as you write.

### 1.1.3.6 **1.3.1**

Bug fixes. Monospaced font now rendering correctly.

Deleting text with backspace have strange effects on text layout. That is, the place where a sentence is broken to the right and moved down to the next line keeps moving around while deleting text, in some completely different paragraph! This is entirely handled by `JTextPane`. I have tried to find a way to intercept the delete key and handle delete myself, but I have not been successful in finding a way to do that if it is even possible. Continuing writing new text after deleting text seems to restore the layout. This oddity has no effect on the final text, it is just the layout while editing that is affected. You will also only see this if you write paragraphs as one block of text that wraps around into multiple lines without pressing return until the end of the paragraph.

### 1.1.3.7 **1.3**

Made big changes to the editor, finally making it into what I want, with some markdown formatting as you write, and far more configuration in settings dialog, which have also been redone.

Bug fixes.

### 1.1.3.8 **1.2.10**

Added support for `&lt;`, `&gt;`, and `&amp;`.

### 1.1.3.9 **1.2.9**

Added markdown file reading feature by allowing markdown files to be dropped on the editor in preview mode, in which case the dropped file will be formatted and displayed without changing the content of the editor. Exiting preview and doing a preview again will again preview the editor content.

### 1.1.3.10 **1.2.8**

Headings can now **not** be more than one line (not include LF/CRLF). Before they were treated like paragraphs. This to be more compatible with other Markdown tools and Markdown documents.

### 1.1.3.11 **1.2.7**

Added settings for specifying top, bottom, left, and right margins in editor. Please note that I've been a bit lazy here. The sizes are in pixels, not characters/lines!

### 1.1.3.12 **1.2.6**

Added the new `.mddoc` format, which makes command line usage easier, but it is also supported by the maven plugin and the library has a utility that completely handles this format.

Added a Java Swing based editor for editing markdown with support.

### 1.1.3.13 **1.2.5**

Added `parserOptions` now used by JavadocParser to markdown parse javadoc comments if `markdownJavadoc=true` is provided. The Parser API is thus also updated to take a Properties object for the parser options.

### 1.1.3.14 **1.2.4**

Added `makeFileLinksRelativeTo` option for HTMLGenerator and MarkdownGenerator mostly to be able to manipulate *file*: references to images in the generated result so that the image paths still work in source when editing with a markdown tool and is still correct when generated to a different path.

### 1.1.3.15 **1.2.3**

If image paths are not absolute and not http referenced then they are now looked for relative to the source markdown file first, and then they are looked for relative to the result file as before. This makes it easier to generate a big document for a whole project containing several subproject with local markdown documents and referenced images. The image reference can still be relative to the subproject local markdown file.

### 1.1.3.16 **1.2.2**

Added support for *to be able to indent text. This is one more exception to no html passthrough.*

## 1.1.4 **How markdown is MarkdownDoc ?**

Well, it implements the "specification" as documented on [daringfireball.net](http://daringfireball.net). This specification however is not extremely exact so there might be some differences.

The known (and intentional) differences are:

- No HTML pass-through! Well, there is a small exception to that. HTML comments are passed along. Mostly because there is no markdown comment format and I wanted to be able to put comments in my documents. "&nbsp;" is also passed through to create indents that are not code blocks. The reason for no HTML pass-through is that MarkdownDoc takes it directly from markdown to PDF without any HTML rendering in between. The main purpose of this tool is to write documentation not generate HTML sites.
- Escaping with '\'. In MarkdownDoc you can escape any character with \ and it will be passed through as is without being acted on if it has markdown meaning.
- No entity encoding of email addresses.
- No multiple block quote levels (as of now).

## 1.2 File specifications

---

With both the maven plugin and the command line execution jar file you can specify a set of files to use as input. These are basically a comma separated list of files, but with the following additions:

`/my/path`

*All files in the directory pointed to by the path.*

`/my/path/**`

*All files in the directory pointed to by the path and sub directories.*

`/my/path/**/regex pattern`

*All files matching the pattern in the directory pointed to by the path and sub directories.*

`/my/path/regex pattern`

*All files matching the pattern in the directory pointed to by the path.*

`/my/path/fileset.fs`

*The above rules are applied to all file specifications in files having the .fs extension. # are comment lines within .fs files.*



## 2 Command Line

### 2.1 General

---

MarkdownDoc can be run using `java -jar markdowndoc-cmd-line-n.n.n-exec.jar`. If you just run it without any arguments you get the following:

```
Usage: java -jar markdowndoc-cmd-line-n.n[.n].exec.jar <generator> --help
      or
      java -jar markdowndoc-cmd-line-n.n[.n].exec.jar <generator> <fileSpec> --
<generator option> ...
      or
      java -jar markdowndoc-cmd-line-n.n[.n].exec.jar <generator> <fileSpec>
parserOptions:<parserOptions> -<generator option> ...
      or
      java -jar markdowndoc-cmd-line-n.n[.n].exec.jar <path to a .mddoc file>
```

The last usage example requires an `.mddoc` file. See '*The mddoc file type*' (section 5) for more information on this file type.

What the generator options are depends on the specified generator.

The `markdowndoc-cmd-line-n.n.n-exec.jar` is a jar generated to contain all dependencies in the same jar, making it easy to execute with `java -jar`.

#### generator

This should be either *pdf*, *html*, or *md*.

#### fileSpec

This is a comma separated list of paths relative to the current directory. The filename part of the path can contain regular expressions and the directory part of the path can specify `.../.../...` to mean any levels of subdirectories.

Example: `root//docs/..md`

#### parserOptions

These are in the format "option=value,...,option=value" with no spaces. Currently there is only one options for the JavadocParser: `markdownJavadoc=true`. When this is specified and java files are part of the input fileSpec then class and method texts are passed to the MarkdownParser instead of being added as text. This can be used in conjunction with a markdown doclet for javadoc.

#### generatorOptions

These depends on the generator being run.

### 2.2 PDF Generator

---

**--resultFile text (Required)** Where to write the result.

**--pageSize text** The pagesize name like LETTER or A4.

**--title text** The title of the document

**--subject text** The subject of the document.

**--keywords text** Meta keywords

**--author text** The author of the document.

**--version text** The version to put on the title page. Must be specified to be rendered!

**--copyright text** The copyright message to put on the title page. Must be specified to be rendered!

**--hideLinks true/false** If true then links are not rendered as link the link text will be rendered as plain text.

**--unorderedListItemPrefix text** What item marking to use for unuredered lists. Default is '- '.

**--firstLineParagraphIndent true/false** If true then the first line of each paragraph is indented. Default is false.

**--backgroundColor text** The background color of the document in "R:G:B" format where each R, G, and B are number 0 - 255.

**--blockquoteColor text** The blockquote color to use in this document in "R:G:B" format where each R, G, and B are number 0 - 255.

**--codeColor text** The code color to use in this document in "R:G:B" format where each R, G, and B are number 0 - 255.

**--generateTOC true/false** This generates table of contents. Default is false!

**--generateTitlePage true/false** This will generate one first page with title, version, author, and copyright. Default is false.

## 2.3 HTML Generator

---

**--resultFile text (Required)** Where to write the result.

**--inlineCSS true/false** If true then the css will be included in the generated HTML.

**--css text** The path to CSS file.

**—makeFileLinksRelativeTo** "path+prefix\_". This affects links and images. When specified the resulting file: URLs in the result will be relative to the path specified by "path" if the absolute path of the URL starts with the specified path. If a plus sign (+) and a prefix path is specified it will be prefixed to the final URL.

## 2.4 Markdown Generator

---

**--resultFile text (Required)** Where to write the result.

**—makeFileLinksRelativeTo** "path+prefix\_". This affects links and images. When specified the resulting file: URLs in the result will be relative to the path specified by "path" if the absolute path of the URL starts with the specified path. If a plus sign (+) and a prefix path is specified it will be prefixed to the final URL.

## 3 Maven Plugin

The maven plugin is rather straight forward. It has 3 sets of configuration structures, one common and one for each generator.

### 3.1 generatorOptions

---

There is a config section that is common to all generators and specifies which generator to run and what input files to include. The following example is from the generation of this manual:

```
<generatorOptions>
  <generator>pdf</generator>
  <inputPaths>
    Docs/parts/H1UserGuide.mdpart,
    Docs/MarkdownDoc.md,
    MavenPlugin/docs/*.md,
    CommandLine/docs/*.md,
    Library/docs/*.md,
    Docs/parts/H1Licenses.mdpart,
    Docs/licenses.md,
    Docs/parts/H1LicenseTexts.mdpart,
    Docs/*.md
  </inputPaths>
  <parserOptions>option=value,...</parserOptions>
</generatorOptions>
```

If the `<inputPaths>...</inputPaths>` section only contains one file of type `.mddoc` then no other parameters need to be specified, not even `<generator>...</generator>`! In this case all information needed to generate final documents resides in the `.mddoc` file. See '*The mddoc file type*' (section 5) for more information on this file type.

The current valid argument for `<generator>...</generator>` are *pdf*, *html*, and *md*.

The input paths are comma separated and are always relative to the root of the maven project. To clarify that, for a multi module maven build it is always the top root with the top pom that is the root even if you start the build at a lower level. This root is resolved by starting at `${basedir}` and going up until the parent directory does not have a pom. I have found no way to let maven tell me this path.

The paths can have wildcards in form of regular expressions for the file names. There is also a special directory name `**` that means any level of subdirectories.

All the input paths are parsed into the same document model that then gets passed to the generator. They are parsed in the order they are specified. When it comes to wildcards it is hard to say which order they will be in. It might differ on different platforms.

If you are wondering about the `.mdpart` extensions above it is just to hide them from GitHub. They are just very small header lines that I inject to the PDF document that I don't want in the main parts of the documentation since they are reused and linked from the README.md in the root.

I'm also using this possibility to generate from multiple sources to put the documentation for each module project in that project.

The `<parserOptions>option=value,...</parserOptions>` are passed to each parser. Currently only the *JavadocParser* has an option: `markdownJavadoc=true`. When this is specified then the class and method descriptions are passed to *MarkdownParser* instead of being added as text. This is useful in conjunction with using a markdown doclet for javadoc.

## 3.2 pdfGeneratorOptions

---

Following is a complete plugin specification with all options specified:

```

<plugin>
  <groupId>se.natusoft.tools.doc.markdowndoc</groupId>
  <artifactId>markdowndoc-maven-plugin</artifactId>
  <version>n.n[.n]</version>

  <executions>
    <execution>
      <id>generate-docs</id>
      <goals>
        <goal>doc</goal>
      </goals>
      <phase>install</phase>
      <configuration>

        <generatorOptions>
          <generator>pdf</generator>
          <inputPaths>
            ...
          </inputPaths>
        </generatorOptions>

        <pdfGeneratorOptions>
          <!--
            The path to the pdf document to produce. Path is relative
            to project root (see comment about root above).
            Required.
          -->
          <resultFile>Docs/MarkdownDoc-User-Guide.pdf</resultFile>

          <!--
            The page size. For example:A4, LETTER
            Optional. Default: A4
          -->
          <pageSize>A4</pageSize>

          <!--
            This will be put in PDF metadata and also rendered on
            title page.
            Required if generateTitlePage is true optional otherwise.
          -->
          <title>MarkdownDoc</title>

          <!--
            This will be put in PDF metadata and also rendered on
            title page.
            Optional.
          -->
          <subject>User Guide</subject>

          <!--
            This will be put in the PDF metadata.
            Optional.
          -->
          <keywords></keywords>

          <!--
            The version of the document. This will be rendered on the
            title page.
            Optional, but recommended if generateTitlePage is true.
          -->
          <version>1.0</version>

          <!--
            The author of the document. This will be put in PDF
            metadata and also be rendered on title page.
            Optional.
          -->
          <author>Tommy Svensson</author>

          <!--
            This will be rendered on the title page.
            Optional
          -->
          <copyright>Copyright © 2012 Natusoft AB</copyright>

          <!--
            If true then links will render as plain text and not be

```

```

        clickable.
        Optional. Default: false
-->
<hideLinks>false</hideLinks>

<!--
    Specify this if you want to change the bullet for
    unordered lists.
    Optional. Default: • (including space after!)
-->
<unorderedListItemPrefix>• </unorderedListItemPrefix>

<!--
    Specify true here to have the first line of each
    paragraph indented.
    Optional. Default: false
-->
<firstLineParagraphIndent>false</firstLineParagraphIndent>

<!--
    Specify in R:G:B format to change the background color
    of the document.
    Optional. Default 255:255:255 (white)
-->
<backgroundColor>255:255:255</backgroundColor>

<!--
    Specify in R:G:B format to change the text color
    of block quotes.
    Optional. Default: 128:128:128 (grey)
-->
<blockquoteColor>128:128:128</blockquoteColor>

<!--
    Specify in R:G:B format to change the text color
    of code blocks.
    Optional. Default: 0:0:0 (black)
-->
<codeColor>0:0:0</codeColor>

<!--
    Set to true to generate a title page.
    Optional. Default: false
-->
<generateTitlePage>true</generateTitlePage>

<!--
    Set to true to generate a table of contents.
    Optional. Default: false
-->
<generateTOC>true</generateTOC>
</pdfGeneratorOptions>

</configuration>
</execution>
</executions>
</plugin>

```

### 3.3 htmlGeneratorOptions

---

Following is a complete plugin specification with all options specified:

```

<plugin>
  <groupId>se.natusoft.tools.doc.markdowndoc</groupId>
  <artifactId>markdowndoc-maven-plugin</artifactId>
  <version>n.n[.n]</version>

  <executions>
    <execution>
      <id>generate-docs</id>
      <goals>
        <goal>doc</goal>
      </goals>
      <phase>install</phase>
      <configuration>

        <generatorOptions>
          <generator>html</generator>
          <inputPaths>
            ...
          </inputPaths>
        </generatorOptions>

        <htmlGeneratorOptions>
          <!--
            The path to the html document to produce. Path is relative
            to project root (see comment about root above).
            Required.
          -->
          <resultFile>Docs/MarkdownDoc-User-Guide.html</resultFile>

          <!--
            If set to true then the specified css will be inlined
            in the generated html document. Otherwise the generated
            html document will reference the specified css.
            Optional. Default: false
          -->
          <inlineCSS>>false</inlineCSS>

          <!--
            The path to the css file for the generated html file.
            Required.
          -->
          <css>css/my.css</css>

          <!--
            This affects links and images. When specified the resulting
            file: URLs in the result will be relative to the path
            "path" if the absolute path of the URL starts with the
            path. If a plus sign (+) and a prefix path is specified it
            prefixed to the final URL.
          -->

          <makeFileLinksRelativeTo>path[+prefix]</makeFileLinksRelativeTo>

        </htmlGeneratorOptions>

      </configuration>
    </execution>
  </executions>
</plugin>

```

specified by

specified

will be

### 3.4 mdGeneratorOptions

Following is a complete plugin specification with all options specified:

```

<plugin>
  <groupId>se.natusoft.tools.doc.markdowndoc</groupId>
  <artifactId>markdowndoc-maven-plugin</artifactId>
  <version>n.n[.n]</version>

  <executions>
    <execution>
      <id>generate-docs</id>
      <goals>
        <goal>doc</goal>
      </goals>
      <phase>install</phase>
      <configuration>

        <generatorOptions>
          <generator>md</generator>
          <inputPaths>
            ...
          </inputPaths>
        </generatorOptions>

        <htmlGeneratorOptions>
          <!--
relative      The path to the markdown document to produce. Path is
               to project root (see comment about root above).
               Required.
          -->
          <resultFile>Docs/MarkdownDoc-User-Guide-Complete.md</resultFile>

          <!--
specified by  This affects links and images. When specified the resulting
specified     file: URLs in the result will be relative to the path
will be       "path" if the absolute path of the URL starts with the
               path. If a plus sign (+) and a prefix path is specified it
               prefixed to the final URL.
          -->

          <makeFileLinksRelativeTo>path[+prefix]</makeFileLinksRelativeTo>

        </htmlGeneratorOptions>

      </configuration>
    </execution>
  </executions>
</plugin>

```



## 4 Library

The library is made up of a document model representing all formats of markdown, parsers and generators. The parsers produce a document model and the generators generate from that model. The document model represents the markdown formats. Thereby there are no HTML pass-through from a markdown document! This tool only deals with markdown, not HTML.

The API docs for the library can be found [here](#).

### 4.1 Usage

---

In package `se.natusoft.doc.markdown.api` there are 3 API classes:

**Options** - This represents options for a generator. It should be seen as a narrow variant of `Object` representing only generator options, but any such. It has one method common to all `public boolean isHelp()`. Implementations should have a default constructor.

**Parser** - This represents a parser.

```
public interface Parser {
    public void parse(Doc document, File parseFile, Properties parserOptions) throws
IOException, ParseException;
}
```

The parser gets passed an already created `Doc` model allowing the document to be built from multiple source files by parsing into the same document.

**Generator** - This represents a generator.

```
public interface Generator {
    public Class getOptionsClass();
    public void generate(Doc document, Options options, File rootDir) throws
IOException, GenerateException;
}
```

`getOptionsClass()` returns the class implementing `Options` and holding all the options for the generator.

`generate(...)` generates the document provided by *document* using the specified *options* and producing the result in whatever *rootDir* relative path is specified in the *options*.

#### 4.1.1 Parsers

##### 4.1.1.1 `se.natusoft.doc.markdown.parser.MarkdownParser`

This parser parses markdown and only markdown! It ignores HTML with the exception of comments.

Example usage:

```
Parser parser = new MarkdownParser();
Doc document = new Doc();
Properties parserOptions = new Properties();
parser.parse(document, parseFile, parserOptions);
```

##### 4.1.1.2 `se.natusoft.doc.markdown.parser.JavadocParser`

This parser parses java source files and extracts class and method declarations and javadoc comment blocks. it produces a document model looking like this (in markdown format):

```

public _class/interface_ __class-name__ extends something [package] {
> class javadoc

__full method declaration__
> method javadoc
_Returns_
> description
_Parameters_
> _param_ - description
_Throws_
> _exception_ - description
_See_
> description

...
}

```

This allows you to include API documentation in your documentation without having to duplicate it. Please note that if `markdownJavadoc=true` parser option have been specified then *class javadoc* and *method javadoc* will not be formatted but passed to the `MarkdownParser` instead.

Example usage:

```

Parser parser = new JavadocParser();
Doc document = new Doc();
Properties parserOptions = new Properties();
parser.parse(document, parseFile, parserOptions);

```

#### 4.1.1.3 **se.natusoft.doc.markdown.parser.ParserProvider**

This is a utility to get a parser based on file extension. ".md", ".markdown", ".mdpart", and ".java" are valid extensions that will return a parser. If the passed file does not have a valid extension null will be returned.

Example usage:

```

Parser parser = ParserProvider.getParserForFile(parseFile);
Doc document = new Doc();
Properties parserOptions = new Properties();
parser.parse(document, parseFile, parserOptions);

```

### 4.1.2 **Generators**

Example usage:

```

public static void main(String[] args) {
    Doc document = new Doc();

    ... parsing of document.

    Generator generator = new [PDF|HTML|Markdown]Generator();

    // I'm using OptionsManager to load the options in this example.
    // If you use maven or ant then those tools will have loaded
    // the options for you and getOptionsClass() is not relevant
    // in that case.
    CommandLineOptionsManager<Options> optMgr =
        new CommandLineOptionsManager<Options>(generator.getOptionsClass());
    Options options = optMgr.loadOptions("--", args);
    if (options.isHelp()) {
        optMgr.printHelpText("--", "", System.out);
    }
    else {
        File rootDir = new File();
        generator.generate(document, options, rootDir);
    }
}

```

Please note that the `CommandLineOptionsMangager` used in the example is part of the `OptionsManager` tool also by me. Available at [github.com/tombensve/OptionsManager](https://github.com/tombensve/OptionsManager).

#### 4.1.2.1 **se.natusoft.doc.markdown.generator.PDFGenerator**

This generator produces a PDF document.

#### 4.1.2.2 **se.natusoft.doc.markdown.generator.HTMLGenerator**

This generator produces an HTML document.

#### 4.1.2.3 **se.natusoft.doc.markdown.generator.MarkdownGenerator**

This generator produces a Markdown document. So why would we want to generate markdown ? Well, it became needed after I added the JavadocParser. Now I can have both markdown and java files as input and the PDF and HTML files contained the whole result including the javadoc information. The original markdown document however does not have the javadoc parts, and this markdown document is read as is on github and will then not be complete. Therefore I added this generator and moved my real source document into docs/src and also generate a markdown version into docs that will be as complete as the pdf and html version.

#### 4.1.2.4 **se.natusoft.doc.markdown.util.MDDocFileHandler**

This is a class with one static method that completely handles the *.mddoc* format.

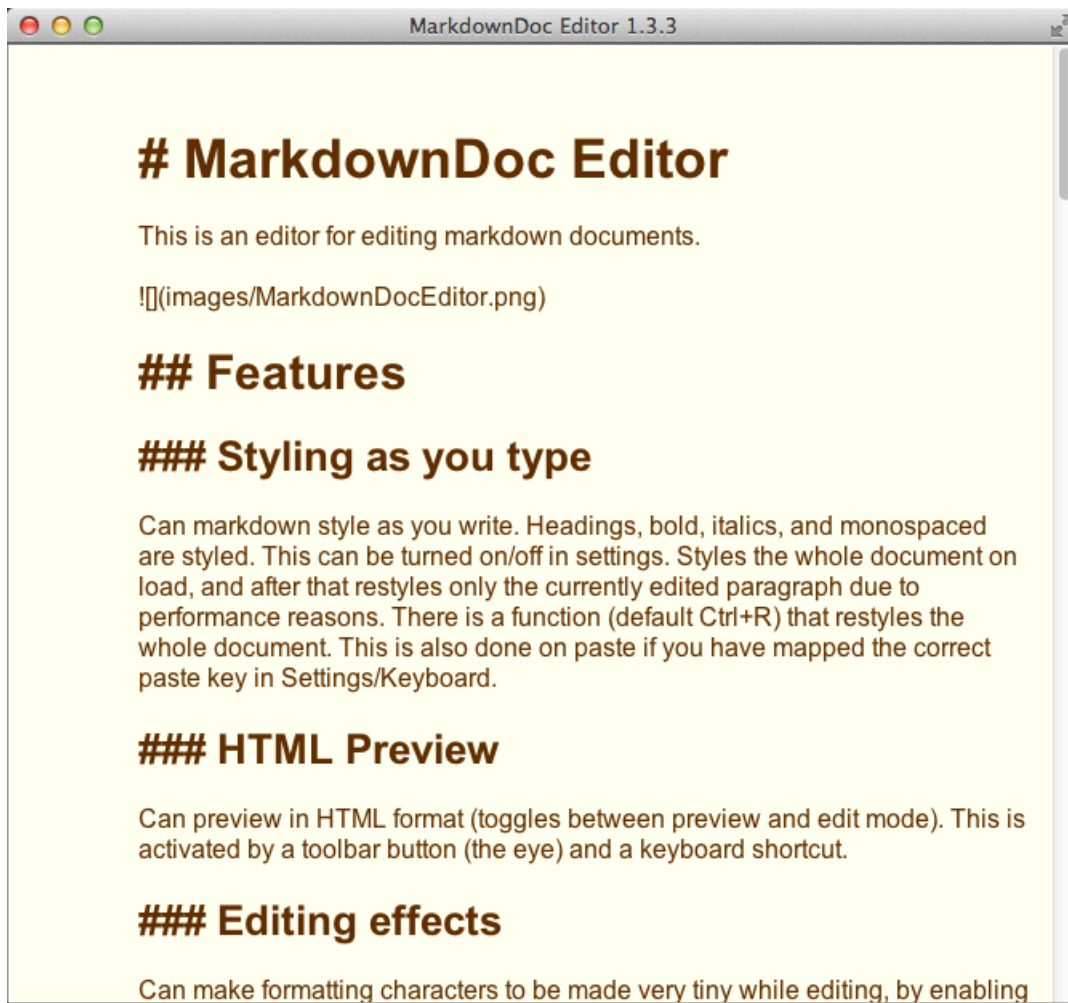
Usage:

```
MDDocFileHandler.execute("<path to .mddoc file>");
```

This will generate all output formats as specified in the *.mddoc* file.

## 5 MarkdownDoc Editor

This is an editor for editing markdown documents.



### 5.1 Features

---

#### 5.1.1 Styling as you type

Can markdown style as you write. Headings, bold, italics, and monospaced are styled. This can be turned on/off in settings. Styles the whole document on load, and after that restyles only the currently edited paragraph due to performance reasons. There is a function (default Ctrl+R) that restyles the whole document. This is also done on paste if you have mapped the correct paste key in Settings/Keyboard.

#### 5.1.2 HTML Preview

Can preview in HTML format (toggles between preview and edit mode). This is activated by a toolbar button (the eye) and a keyboard shortcut.

#### 5.1.3 Editing effects

Can make formatting characters to be made very tiny while editing, by enabling a settings option. Try it to see the effect!

#### 5.1.4 Generate PDF & HTML

Can generate both PDF and HTML from the editor window. Use toolbar button or keyboard shortcut.

### 5.1.5 Configurable

The settings dialog allows you to configure almost anything/everything:

- All keyboard shortcuts.
  - Don't write the keyboard shortcut in text, just press the keyboard shortcut you want to set.
  - Configured keyboard values are stored in their string representation and matched as strings.
    - This means that the code does not have to do a humongous if statement set for each possible alternative.
    - This also means that due to differences in java implementations and versions the string representation might be Ctrl+K or ^+K. So if you change java version you might also have to update keyboard mappings in settings.
- Margins.
- Editor font.
- Monospaced font.
- Preview font.
- Font sizes.
- Background color.
- Text color.
- Toolbar variant to use.

### 5.1.6 Load file by drag & drop

Instead of using the GUI open dialog you can just drag and drop a file in the editor to edit it.

### 5.1.7 Special preview drag & drop feature

While in preview mode, drag and drop a markdown file on the preview window to have it formatted and displayed. This does not affect the edit buffer in any way. Exiting preview mode will bring you back to whatever you have in the editor, and previewing again will preview the editor content.

But by just opening an empty editor and entering a blank preview you can quickly read multiple markdown documents formatted by just dropping them on the window.

### 5.1.8 Mac OS X Fullscreen support

When you run this editor on a Mac with Lion+ you will get a litte double arrow in the right corner of the window titlebar, which will bring upp the editor window in fullscreen.

## 5.2 Running

---

Can be run with `java -jar` or double clicked on. If you are using Windows 7 or 8 take a look at this page:

<http://johann.ioefflmann.net/en/software/jarfix/index.html> (<http://johann.ioefflmann.net/en/software/jarfix/index.html>).

The executable jar have the following name: `MarkdownDocEditor-n.n.n-App.jar`

One or more files can be specified as arguments.

## 5.3 Requirements

---

This requires Java 7+!

## 5.4 Functions

---

Do note that since all keyboard actions can be configured in settings this documents the default keyboard settings.

Also note that the defaults are adapted for Windows and Linux. On a mac you might want to change Ctrl to the Cmd key instead.

	Keyboard default	Available in toolbar
Save file	Ctrl+S	Yes
Open file	Ctrl+O	Yes
Open new window	Ctrl+N	Yes
Insert heading	Ctrl+T	Yes
Insert bold	Ctrl+B	Yes
Insert italics	Ctrl+I	Yes
Insert list	Ctrl+L	Yes
Insert quote	Ctrl+Q	Yes
Insert image	Ctrl+M	Yes
Insert link	Ctrl+N	Yes
Preview	Ctrl+F	Yes
Generate PDF	Ctrl+P	Yes
Generate HTML	Ctrl+H	Yes
Settings	Ctrl+E	Yes
Restyle document	Ctrl+R	No
Restyle on paste (*)	Ctrl+V	No

(\*) This can be disabled by setting the key to anything other than the paste key.

## 5.5 If you're on a Mac

If you are on a Mac you might want to change the keyboard mappings to use Cmd rather than Ctrl. Do note however that Cmd+H and Cmd+Q are really nasty on Mac OS X! Since these keys immediately kills the app these keys are impossible to set in the first place, but you will loose other unsaved settings when you try.

## 5.6 Currently Missing

Fancy functions like search and replace.

Undo capability.

## 5.7 Bugs

### 5.7.1 By me

Only images with absolute path (even http: urls) are rendered in preview. Not sure I can fix this since the preview is generated in memory. I don't really know what the JTextPane sees links relative to then. Possibly if I can resolve the full path to a relative image using the same scheme as used in the PDFGenerator.

### 5.7.2 By Oracle

- This editor uses the standard Swing component JTextPane. This is unfortunately not an optimal component. Specially for styling it gets slow for large documents. In earlier versions of Java 7 this component had a word wrap problem when deleting text either using backspace or cutting text. In that case it rerendered the text screwing up the format until new text was entered again. *As of Java 8 this bug is fixed*, but other new bugs have been added. They are however smaller and don't occur so often.
- Sometimes when the JTextPane is opened the pane will not render at all! Just increase the width of the window until text appears. Then save so that the window size for that file will be remembered. I have one and only one document for which this happens and I cannot tell what it is that causes the problem. This could be a mac only problem.

## 6 The mddoc file type

There is a special file type that describes a complete document in any or all of the 3 output formats. It has the extension of *.mddoc*. It is really a properties file with *key: value* entries.

A path to an *.mddoc* file can be specified as only argument to command line variant, or as only file in `<inputPaths>...</inputPaths>` section in maven plugin (no other options/parameters are needed then) to produce output documents as described by the *.mddoc* file.

### 6.1 .mddoc format (myfile.mddoc)

---

```

# --- Generators to run ---
generate.pdf: true
generate.html: true
generate.markdown: true

# A comma separated list of paths to sources. A .fs file can also be
# specified as an input file in which case it is read for a further
# set of files to parse. The order of the specified files are important.
inputPaths: docs/intro.md,docs/install.md,docs/usage.md,docs/appendix.md

# --- PDF ---

# The name of the file to produce.
pdf.resultFile: MyDoc.pdf

# The page size. For example:A4, LETTER Optional. Default: A4
pdf.pageSize: A4

# Set to true to generate a title page. Optional. Default: false
generateTitlePage: true/false

# Set to true to generate a table of contents. Optional. Default: false
pdf.generateTOC: true/false

# This will be put in PDF metadata and also rendered on title page.
# Required if generateTitlePage is true optional otherwise.
pdf.title: MyDoc

# This will be put in PDF metadata and also rendered on title page.
# Optional.
pdf.subject: User Guide

# This will be put in the PDF metadata. Optional.
pdf.keywords: Markdown PDF

# The version of the document. This will be rendered on the title page.
# Optional, but recommended if generateTitlePage is true.
pdf.version: 1.0

# The author of the document. This will be put in PDF metadata and also
# be rendered on title page. Optional.
pdf.author: Your Name

# This will be rendered on the title page. Optional.
pdf.copyright: Copyright ?? 2013 Me Myself and I

# If true then links will render as plain text and not be clickable.
# Optional. Default: false
pdf.hideLinks: true/false

# Specify this if you want to change the bullet for unordered lists.
# Optional. Default: ??? (including space after!)
pdf.unorderedListItemPrefix: *

# Specify true here to have the first line of each paragraph indented.
# Optional. Default: false
pdf.firstLineParagraphIndent: true

# Specify in R:G:B format to change the background color of the document.
# Optional. Default 255:255:255 (white)
pdf.backgroundColor: 255:255:255

# Specify in R:G:B format to change the text color of block quotes.
# Optional. Default: 128:128:128 (grey)
pdf.blockQuoteColor: 128:128:128

# Specify in R:G:B format to change the text color of code blocks.
# Optional. Default: 0:0:0 (black)
pdf.codeColor: 0:0:0

# --- HTML ---

# The name of the file to produce.
html.resultFile: MyDoc.html

# The path to the css file for the generated html file. Required.
html.css: css/my.css

```



```
# If set to true then the specified css will be inlined in the generated html
# document. Otherwise the generated html document will reference the specified
# css. Optional. Default: false
html.inlineCSS: true/false
```

```
# This affects links and images. When specified the resulting file: URLs in the
# result will be relative to the path specified by "path" if the absolute path
# of the URL starts with the specified path. If a plus sign (+) and a prefix
# path is specified it will be prefixed to the final URL. Optional.
html.makeFileLinksRelativeTo: path[+prefix]
```

```
# --- Markdown ---
```

```
# The name of the file to produce.
markdown.resultFile: MyDoc.md
```

```
# This affects links and images. When specified the resulting file: URLs in the
# result will be relative to the path specified by "path" if the absolute path
# of the URL starts with the specified path. If a plus sign (+) and a prefix
# path is specified it will be prefixed to the final URL. Optional.
markdown.makeFileLinksRelativeTo: path[+prefix]
```

## 7 Simple Markdown Reference

### 7.1 Headings

---

```
# Heading level 1
## Heading level 2
...
##### Heading level 6
```

### 7.2 Paragraphs

---

An empty line marks the end of a paragraph.

```
Paragraph 1 ...
More text in paragraph 1.

Paragraph 2 ...
```

Paragraph 1 ... More text in paragraph 1.

Paragraph 2 ...

### 7.3 Italics

---

```
_This is in italics_
```

This *\*is also italics\** but can't start a line with *\** since it will be treated as list.

*This is in italics*

This *is also italics* but can't start a line with *\** since it will be treated as list.

### 7.4 Bold

---

```
__This text is bold.__
```

```
**This text is also bold.**
```

**This text is bold.**

**This text is also bold.**

### 7.5 Blockquote

---

```
> This line is block quoted.
```

*This line is block quoted.*

### 7.6 Lists

---

#### 7.6.1 Unordered lists (\* or -)

```
* item 1
  Also part of item 1.

- item 2
  * item 2.1 (this is indented 4 spaces!)
```

- item 1 Also part of item 1.
- item 2
  - item 2.1 (this is indented 4 spaces!)

### 7.6.2 Ordered list (n.)

```
1. item 1.
2. item 2.
  1. item 2.1
```

1. item 1.
2. item 2.
  1. 1. item 2.1

Please note that the actual numbers does not matter! They could all be "1.!" The items will be enumerated automatically in order no matter what numbers you enter in the source.

## 7.7 Code block (pre formatted with a fixed width font)

---

Each line starting with a tab or 4 spaces are considered belonging to a pre formatted block.

```
This
  is
    a
      preformatted
        block!
```

## 7.8 Horizontal rule

---

Any of:

```
* * *
***
*****...***
- - -
----
-----...----
```

## 7.9 Links

---

```
[This is a link to markdown syntax on
daringfireball.net](http://daringfireball.net/projects/markdown/syntax)
```

This is a link to markdown syntax on [daringfireball.net](http://daringfireball.net)

Short "autolink" version:

```
<http://www.daringfireball.net>
```

<http://www.daringfireball.net>

## 7.10 Images

---

```
![Alt text](/path/to/img.png)
```

```
![Alt text](/path/to/img.png "title")
```

## 7.11 backslash (\)

---

The \ character can be used to escape characters that have markdown meaning. \\ will for example produce \. \\* will produce \*.

## 8 Licenses

### 8.1 Project License

---

Apache Software License version 2.0

### 8.2 Third Party Licenses

---

Apache Software License version 2.0

The following third party products are using this license:

- OptionsManager-2.0.3
- groovy-all-2.3.6

GNU Affero General Public License version v3

The following third party products are using this license:

- itext-pdfa-5.5.1

## 9 License Texts

### 9.1 Apache License version 2.0, January 2004

---

<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

1. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such

Derivative Works in Source or Object form.

2. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
3. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  1. 1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
  2. 2. You must cause any modified files to carry prominent notices stating that You changed the files; and
  3. 3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  4. 4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

4. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
5. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
6. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
7. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

8. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### 9.1.1 **APPENDIX: How to apply the Apache License to your work.**

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

## 9.2 **GNU Affero General Public License version v3**

---

### 9.2.1 **GNU AFFERO GENERAL PUBLIC LICENSE**

Version 3, 19 November 2007

Copyright © 2007 Free Software Foundation, Inc.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### **Preamble**

The GNU Affero General Public License is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, our General Public Licenses are intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.



Developers that use our General Public Licenses protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License which gives you legal permission to copy, distribute and/or modify the software.

A secondary benefit of defending all users' freedom is that improvements made in alternate versions of the program, if they receive widespread use, become available for other developers to incorporate. Many developers of free software are heartened and encouraged by the resulting cooperation. However, in the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.

The GNU Affero General Public License is designed specifically to ensure that, in such cases, the modified source code becomes available to the community. It requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

An older license, called the Affero General Public License and published by Affero, was designed to accomplish similar goals. This is a different license, not a version of the Affero GPL, but Affero has released a new version of the Affero GPL which permits relicensing under this license.

The precise terms and conditions for copying, distribution and modification follow.

## 9.2.2 TERMS AND CONDITIONS

### 9.2.2.1 0. Definitions.

"This License" refers to version 3 of the GNU Affero General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you".

"Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

### 9.2.2.2 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code"

means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

### 9.2.2.3 **2. Basic Permissions.**

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

### 9.2.2.4 **3. Protecting Users' Legal Rights From Anti-Circumvention Law.**

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article

11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and

you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

#### 9.2.2.5 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 9.2.2.6 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section

1. 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

#### 9.2.2.7 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source.

This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

### 9.2.2.8 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to

part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

### 9.2.2.9 **8. Termination.**

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

### 9.2.2.10 **9. Acceptance Not Required for Having Copies.**

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

### 9.2.2.11 **10. Automatic Licensing of Downstream Recipients.**

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

### 9.2.2.12 **11. Patents.**

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

### 9.2.2.13 **12. No Surrender of Others' Freedom.**

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

### 9.2.2.14 **13. Remote Network Interaction; Use with the GNU General Public License.**

Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source for any work covered by version 3 of the GNU General Public License that is incorporated pursuant to the following paragraph.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the work with which it is combined will remain governed by version 3 of the GNU General Public License.

### 9.2.2.15 **14. Revised Versions of this License.**

The Free Software Foundation may publish revised and/or new versions of the GNU Affero 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 Program specifies that a certain numbered version of the GNU Affero General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU Affero General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU Affero General Public License can

be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### 9.2.2.16 **15. Disclaimer of Warranty.**

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### 9.2.2.17 **16. Limitation of Liability.**

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 9.2.2.18 **17. Interpretation of Sections 15 and 16.**

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

### 9.2.3 **How to Apply These Terms to Your New Programs**

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>; <name of author>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as
published by the Free Software Foundation, either version 3 of the
License, or (at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU Affero General Public License for more details.
```

```
You should have received a copy of the GNU Affero General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
```



Also add information on how to contact you by electronic and paper mail.

If your software can interact with users remotely through a computer network, you should also make sure that it provides a way for users to get its source. For example, if your program is a web application, its interface could display a "Source" link that leads users to an archive of the code. There are many ways you could offer source, and different solutions will be better for different programs; see section 13 for the specific requirements.

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU AGPL, see <http://www.gnu.org/licenses/>.