

Regular Expressions

en.wikibooks.org

March 21, 2015

On the 28th of April 2012 the contents of the English as well as German Wikibooks and Wikipedia projects were licensed under Creative Commons Attribution-ShareAlike 3.0 Unported license. A URI to this license is given in the list of figures on page 41. If this document is a derived work from the contents of one of these projects and the content was still licensed by the project under this license at the time of derivation this document has to be licensed under the same, a similar or a compatible license, as stated in section 4b of the license. The list of contributors is included in chapter Contributors on page 39. The licenses GPL, LGPL and GFDL are included in chapter Licenses on page 45, since this book and/or parts of it may or may not be licensed under one or more of these licenses, and thus require inclusion of these licenses. The licenses of the figures are given in the list of figures on page 41. This PDF was generated by the \LaTeX typesetting software. The \LaTeX source code is included as an attachment (`source.7z.txt`) in this PDF file. To extract the source from the PDF file, you can use the `pdfdetach` tool including in the `poppler` suite, or the <http://www.pdfplabs.com/tools/pdftk-the-pdf-toolkit/> utility. Some PDF viewers may also let you save the attachment to a file. After extracting it from the PDF file you have to rename it to `source.7z`. To uncompress the resulting archive we recommend the use of <http://www.7-zip.org/>. The \LaTeX source itself was generated by a program written by Dirk Hünninger, which is freely available under an open source license from http://de.wikibooks.org/wiki/Benutzer:Dirk_Huenniger/wb2pdf.

Contents

1	Introduction	3
2	Syntaxes	7
3	Simple Regular Expressions	9
3.1	Examples	10
3.2	Use in Tools	10
4	Basic Regular Expressions	11
4.1	Use in Tools	11
5	Perl-Compatible Regular Expressions	13
5.1	Examples	13
5.2	Use in Tools	15
5.3	See also	16
5.4	Links	16
6	POSIX Basic Regular Expressions	17
6.1	External links	20
6.2	Use in Tools	20
6.3	Links	20
7	POSIX-Extended Regular Expressions	21
7.1	Use in Tools	23
7.2	Links	23
8	Non-POSIX Basic Regular Expressions	25
8.1	Use in Tools	25
9	Emacs Regular Expressions	27
9.1	Use in Tools	27
9.2	Links	27
10	Shell Regular Expressions	29
10.1	Use in Tools	29
11	Implementation	31
12	Glossary	33
13	Links	37
13.1	See also	37

13.2 External Links	37
14 Contributors	39
List of Figures	41
15 Licenses	45
15.1 GNU GENERAL PUBLIC LICENSE	45
15.2 GNU Free Documentation License	46
15.3 GNU Lesser General Public License	47

1 Introduction

1.0.1 What is a regular expression?

A **regular expression** is a method of representing a string matching pattern. Regular expressions enable strings that match a particular pattern within textual data records to be located and modified and they are often used within utility programs and programming languages that manipulate textual data. Regular expressions are extremely powerful.

1.0.2 Example applications

Various software applications use regular expressions to locate, select or modify particular sections text. For example, a regular expression could be used to:

- replace the word "snake" with the word "serpent" throughout an entire piece of text
- locate pieces of text containing the words "fox" and "sheep" on the same line

1.0.3 Regular expression components

Regular expressions are made up of three types of components:

- anchors used to specify the position of the pattern in relation to a line of text.
- character sets used to match one or more characters in a single position.
- modifiers used to specify how many times a character set is repeated.

1.0.4 Syntax varies across application programs

The syntax of regular expressions varies across application programs. For example the shell uses a limited form of regular expression called shell regular expressions for filename substitution, whereas AWK¹ uses a superset of extended regular expressions syntax.

1.0.5 Supporting Software

Regular expressions are supported by various software tools, including command line tools, plain text editors and programming languages. Most of these tools are available for various computing platforms, including Linux, Windows and Mac OS X. The tools use slightly different syntax styles. Let's look at some notable ones.

¹ <http://en.wikibooks.org/wiki/AWK>

The tools that use regular expressions are enlisted below:

Command line tools	Plain text editors	Programming languages
grep	ed	.NET
egrep	vi	Awk
sed	Emacs	Java
		JavaScript
		Perl
		PHP
		Python
		Ruby
		Tcl

A regular expression can be considered to be a little computer program that finds or isolates a subset of a larger set of text. In the same way that an ordinary computer program needs a computer to execute it, a regular expression needs a software application to interpret it — to give it meaning.

For example, a regular expression can be used to tell an editor to find the next occurrence of the word "Chapter" followed by several spaces and digits. Or you can use a regular expression to tell the UNIX grep command to show only those lines of a file that contain the word "Wiki" followed by either the word "Books" or the word-fragment "pedia". We will discuss the exact syntax of such regular expressions in the next chapter.

Category:Regular Expressions²

pt:Expressões regulares/Introdução³

² <http://en.wikibooks.org/wiki/Category%3ARegular%20Expressions>

³ <http://pt.wikibooks.org/wiki/Express%C3%B5es%20regulares%2FIntrodu%C3%A7%C3%A3o>

2 Syntaxes

There are several variants of regular expressions. These variants differ not only in their concrete syntax but also in their capabilities. Individual tools that support regular expressions also have their own peculiarities.

- `../Simple Regular Expressions`¹ - widely used for backwards compatibility, but deprecated on POSIX-compliant systems.
- `../Basic Regular Expressions`² - used by some Unix shell tools
- `../Perl-Compatible Regular Expressions`³ - used by Perl and some application programs
- `../POSIX Basic Regular Expressions`⁴ - provides extensions for consistency between utility programs. These extensions are not supported by some traditional implementations of Unix tools.
- `../POSIX-Extended Regular Expressions`⁵ - may be supported by some Unix utilities via the `-E` command line switch
- `../Non-POSIX Basic Regular Expressions`⁶ - provides additional character classes not supported by POSIX
- `../Emacs Regular Expressions`⁷ - used by the Emacs editor
- `../Shell Regular Expressions`⁸ - a limited form of regular expression used for pattern matching and filename substitution

2.0.6 Greedy expressions

Quantifiers such as `*` and `+` match as much as they can: they are greedy. For some uses, their greediness does not fit. For example, let us assume you want to find the first string enclosed in quotation marks, in the following text:

These words include *"cat"*, *"mat"*, and *"pat"*.

The pattern `".*"` matches the italicized part of the text below, that is, *"cat"*, *"mat"*, and *"pat"* instead of the desired *"cat"*:

These words include *"cat"*, *"mat"*, and *"pat"*.

To fix this, some flavours of regular expressions provide non-greedy operators such as `*?`, `+?`, and `}?.` In PHP, adding a `"U"` at the end of the regexp makes the quantifier non-greedy,

1 Chapter 2.0.7 on page 8
2 Chapter 3.2 on page 10
3 Chapter 4.1 on page 11
4 Chapter 5.4 on page 16
5 Chapter 6.3 on page 20
6 Chapter 7.2 on page 23
7 Chapter 8.1 on page 25
8 Chapter 9.2 on page 27

as in `/".*"/U`. In flavours that support neither of the two options, you can specify what is *not* to be matched, as in `("^[^"]*")` to fix the discussed example. However, when dealing with bracketed expressions, `(\[\[[^\]]*\]\])` fails to match on `A B C [[D E] F G]`.

2.0.7 Comparison table

A comparison table or matrix that shows which features or flavors of regular expressions are available in which tool or programming language is available from regular-expressions.info⁹.

10

9 <http://www.regular-expressions.info/refflavors.html>

10 <http://en.wikibooks.org/wiki/Category%3ARegular%20Expressions>

3 Simple Regular Expressions

The **Simple Regular Expression** syntax is widely used on Unix based systems for the purposes of backwards compatibility. Most regular-expression-aware Unix utilities, such as *grep* and *sed*, use it by default while providing support for extended regular expressions with command line arguments (see below). This syntax is deprecated on POSIX compliant systems and should not be used by new utilities.

When simple regular expression syntax is being used, most characters, except metacharacters are treated as literal characters and match only themselves (for example, "a" matches "a", "(bc" matches "(bc", etc).

Operators	
Operator	Effect
.	The dot operator matches any single character.
[]	boxes enable a single character to be matched against character lists or character ranges.
[^]	A complement box enables a single character not within in a character list or character range to be matched.
^	A caret anchor matches the start of the line (or any line, when applied in multiline mode)
\$	A dollar anchor matches the end of the line (or any line, when applied in multiline mode)
()	parentheses are used to define a marked subexpression. The matched text section can be recalled at a later time.
\n	Where <i>n</i> is a digit from 1 to 9; matches what the <i>n</i> th marked subexpression matched. This irregular construct has not been adopted in the extended regular expression syntax.
*	<p>A single character expression followed by "*" matches zero or more copies of the expression. For example, "ab*c" matches "ac", "abc", "abbbc" etc. "[xyz]*" matches "", "x", "y", "zx", "zyx", and so on.</p> <ul style="list-style-type: none"> • \n*, where <i>n</i> is a digit from 1 to 9, matches zero or more iterations of what the <i>n</i> th marked subexpression matched. For example, "(a.)c\1*" matches "abcab" and "abcabab" but not "abcac". • An expression enclosed in "(" and ")" followed by "*" is deemed to be invalid. In some cases (e.g. /usr/bin/xpg4/grep of SunOS 5.8), it matches zero or more iterations of the string that the enclosed expression matches. In other cases (e.g. /usr/bin/grep of SunOS 5.8), it matches what the enclosed expression matches, followed by a literal "*".

3.1 Examples

Examples:

- `^[hc]at`
 - Matches *hat* and *cat* but only at the beginning of a line.
- `[hc]at$`
 - Matches *hat* and *cat* but only at the end of a line.

3.2 Use in Tools

Tools and languages that utilize this regular expression syntax include:

- Grep¹
- sed²

¹ <http://en.wikibooks.org/wiki/Grep>

² <http://en.wikibooks.org/wiki/sed>

4 Basic Regular Expressions

Basic Regular Expressions : Note that particular implementations of regular expressions interpret the backslash symbol differently in front of some of the metacharacters. For example, `grep` and `perl` interpret *unbackslashed* parentheses and vertical bars as metacharacters, reserving the backslashed versions to mean the literal characters themselves. Old versions of `grep` did not support the pipe alternation operator.

Operators	
Operator	Effect
.	The dot operator matches any single character.
[]	boxes enable a single character to be matched against a character lists or character range.
[^]	A compliment box enables a single character not within in a character list or character range to be matched.
*	An asterisk specifies zero or more characters to match.
^	The caret anchor matches the beginning of the line
\$	The dollar anchor matches the end of the line

Examples:	
Example	Match
".at"	any three-character string like <i>hat</i> , <i>cat</i> or <i>bat</i>
"[hc]at"	<i>hat</i> and <i>cat</i>
"[^b]at"	all the matched strings from the regex ".at" except <i>bat</i>
"^[hc]at"	<i>hat</i> and <i>cat</i> but only at the beginning of a line
"[hc]at\$"	<i>hat</i> and <i>cat</i> but only at the end of a line

Since many ranges of characters depends on the chosen locale setting (e.g., in some settings letters are organized as `abc..yzABC..YZ` while in some others as `aAbBcC..yYzZ`).

The `../Posix Basic Regular Expressions`¹ syntax provided extensions for consistency between utility programs such as `grep`, `sed` and `awk`. These extensions are not supported by some traditional implementations of Unix tools.

4.1 Use in Tools

Tools and languages that utilize this regular expression syntax include: TBD

¹ Chapter 5.4 on page 16

5 Perl-Compatible Regular Expressions

Perl has a richer and more predictable syntax than even the `../POSIX Extended Regular Expressions`¹ syntax. An example of its predictability is that `\` always quotes a non-alphanumeric character. An example of something that is possible to specify with Perl but not POSIX is whether part of the match wanted to be greedy or not. For instance in the pattern `/a.*b/`, the `.*` will match as much as it can, while in the pattern `/a.*?b/`, `.*`? will match as little. So given the string "a bad dab", the first pattern will match the whole string, and the second will only match "a b".

For these reasons, many other utilities and applications have adopted syntaxes that look a lot like Perl's. For example, Java, Ruby, Python, PHP, `exim`, `BBEdit`, and even Microsoft's `.NET Framework` all use regular expression syntax similar to that used in `perl`. Not all "Perl-compatible" regular expression implementations are identical, and many implement only a subset of Perl's features.

5.1 Examples

Conventions used in the examples: The character 'm' is not always required to specify a `perl` match operation. For example, `m/[^abc]/` could also be rendered as `/[^abc]/`. The 'm' is only necessary if the user wishes to specify a match operation without using a forward-slash as the regex delimiter. Sometimes it is useful to specify an alternate regex delimiter in order to avoid "delimiter collision". See 'perldoc perlre²' for more details.

```
metacharacter(s) ;; the metacharacters column specifies the regex syntax
being demonstrated
=~ m//           ;; indicates a regex match operation in perl
=~ s//          ;; indicates a regex substitution operation in perl
```

In the table heading below, "M-c" stands for "Metacharacters".

M-c	Description	Example
.	Normally matches any character except a newline. Within square brackets the dot is literal.	All the if statements return a TRUE value. <pre>if ("Hello World\n" =~ m/...../) { print "Yep"; # Has length >= 5\n"; }</pre>

1 <http://en.wikibooks.org/wiki/..%2FPOSIX%20Extended%20Regular%20Expressions%2F>
2 <http://perldoc.perl.org/perlre.html>

M-c	Description	Example
		All the if statements return a TRUE value.
()	Groups a series of pattern elements to a single element. When you match a pattern within parentheses, you can use any of \$1, \$2, ... later to refer to the previously matched pattern.	<pre>if ("Hello World\n" =~ m/(H..)(o..)/) { print "We matched '\$1' and '\$2'\n"; } Output: We matched 'Hel' and 'o W';</pre>
+	Matches the preceding pattern element one or more times.	<pre>if ("Hello World\n" =~ m/l+/) { print "One or more 'l'\n"; }</pre>
?	Matches the preceding pattern element zero or one times.	<pre>if ("Hello World\n" =~ m/H.?e/) { print "There is an 'H' and a 'e' separated by "; print "0-1 characters (Ex: He Hoe)\n"; }</pre>
?	Modifies the *, +, or {M,N}'d regexp that comes before to match as few times as possible.	<pre>if ("Hello World\n" =~ m/(l.+?o)/) { print "Yep"; # The non-greedy match with 'l' followed # by one or more characters is 'llo' rather than 'llo wo'. }</pre>
*	Matches the preceding pattern element zero or more times.	<pre>if ("Hello World\n" =~ m/el*o/) { print "There is an 'e' followed by zero to many "; print "'l' followed by 'o' (eo, elo, ello, ello)\n"; }</pre>
{M,N}	Denotes the minimum M and the maximum N match count.	<pre>if ("Hello World\n" =~ m/l{1,2}/) { print "There is a substring with at least 1 "; print "and at most 2 l's in the string\n"; }</pre>
[...]	Denotes a set of possible character matches.	<pre>if ("Hello World\n" =~ m/[aeiou]+)/ { print "Yep"; # Contains one or more vowels }</pre>
	Separates alternate possibilities.	<pre>if ("Hello World\n" =~ m/(Hello Hi Pogo)/) { print "At least one of Hello, Hi, or Pogo is "; print "contained in the string.\n"; }</pre>
\b	Matches a word boundary.	<pre>if ("Hello World\n" =~ m/llo\b/) { print "There is a word that ends with 'llo'\n"; }</pre>
\w	Matches an alphanumeric character, including "_".	<pre>if ("Hello World\n" =~ m/\w/) { print "There is at least one alphanumeric "; print "character in the string (A-Z, a-z, 0-9, _)\n"; }</pre>
\W	Matches a non-alphanumeric character, excluding "_".	<pre>if ("Hello World\n" =~ m/\W/) { print "The space between Hello and "; print "World is not alphanumeric\n"; }</pre>

M-c	Description	Example
\s	Matches a whitespace character (space, tab, newline, form feed)	All the if statements return a TRUE value. <pre>if ("Hello World\n" =~ m/\s.*\s/) { print "There are TWO whitespace characters, which may"; print " be separated by other characters, in the string."; }</pre>
\S	Matches anything BUT a whitespace.	<pre>if ("Hello World\n" =~ m/\S.*\S/) { print "Contains two non-whitespace characters " . "separated by zero or more characters."; }</pre>
\d	Matches a digit, same as [0-9].	<pre>if ("99 bottles of beer on the wall." =~ m/(\d+)/) { print "\$1 is the first number in the string'\n"; }</pre>
\D	Matches a non-digit.	<pre>if ("Hello World\n" =~ m/\D/) { print "There is at least one character in the string"; print " that is not a digit.\n"; }</pre>
^	Matches the beginning of a line or string.	<pre>if ("Hello World\n" =~ m/^He/) { print "Starts with the characters 'He'\n"; }</pre>
\$	Matches the end of a line or string.	<pre>if ("Hello World\n" =~ m/rld\$/) { print "Is a line or string "; print "that ends with 'rld'\n"; }</pre>
\A	Matches the beginning of a string (but not an internal line).	<pre>if ("Hello\nWorld\n" =~ m/\AH/) { print "Yep"; # The string starts with 'H'. }</pre>
\Z	Matches the end of a string (but not an internal line).	<pre>if ("Hello\nWorld\n"; =~ m/d\n\Z/) { print "Yep"; # Ends with 'd\n'\n"; }</pre>
[^...]	Matches every character except the ones inside brackets.	<pre>if ("Hello World\n" =~ m/[^abc]/) { print "Yep"; # Contains a character other than a, b, and c. }</pre>

5.2 Use in Tools

Tools and languages that utilize Perl regular expression syntax include:

- Java³
- leafnode
- Perl⁴

³ <http://en.wikibooks.org/wiki/Java%20Programming>

⁴ <http://en.wikibooks.org/wiki/Perl%20Programming>

- Python⁵
- PHP⁶

5.3 See also

- Perl Programming/Regular Expressions Reference⁷

5.4 Links

- Perl regular expressions⁸ at perl.org
- Perl Compatible Regular Expressions library⁹ at pcre.org
- Perl Regular Expression Syntax¹⁰ at boost.org
- W:Regular_expressions#Standard_Perl¹¹

5 <http://en.wikibooks.org/wiki/Python>

6 <http://en.wikibooks.org/wiki/PHP>

7 <http://en.wikibooks.org/wiki/Perl%20Programming%2FRegular%20Expressions%20Reference>

8 <http://perldoc.perl.org/perlre.html>

9 <http://www.pcre.org/>

10 http://www.boost.org/doc/libs/1_34_1/libs/regex/doc/syntax_perl.html

11 http://en.wikipedia.org/wiki/Regular_expressions%23Standard_Perl

6 POSIX Basic Regular Expressions

The **POSIX Basic Regular Expression** (BRE) syntax provided extensions to achieve consistency between utility programs such as `grep`, `sed` and `awk`. These extensions are not supported by some traditional implementations of Unix tools.

6.0.1 History

Traditional Unix regular expression syntax followed common conventions that often differed from tool to tool. The POSIX Basic Regular Expressions syntax was developed by the IEEE, together with an extended variant called `Extended Regular Expression`¹ syntax. These standards were designed mostly to provide backward compatibility with the traditional `../Simple Regular Expressions`/² syntax, providing a common standard which has since been adopted as the default syntax of many Unix regular expression tools.

6.0.2 Syntax

In POSIX Basic Regular Expression syntax, most characters are treated as literals — they match only themselves (e.g., `a` matches `"a"`). The exceptions, listed below, are called *metacharacters* or *metasequences*.

Metacharacter	Description
.	Matches any single character (many applications exclude newlines, and exactly which characters are considered newlines is flavor, character encoding, and platform specific, but it is safe to assume that the line feed character is included). Within POSIX bracket expressions, the dot character matches a literal dot. For example, <code>a.c</code> matches <code>"abc"</code> , etc., but <code>[a.c]</code> matches only <code>"a"</code> , <code>"</code> , or <code>"c"</code> .

¹ <http://en.wikibooks.org/wiki/ERE>

² Chapter 2.0.7 on page 8

Metacharacter	Description
[]	A bracket expression. Matches a single character that is contained within the brackets. For example, [abc] matches "a", "b", or "c", and [a-z] specifies a range which matches any lowercase letter from "a" to "z". These forms can be mixed: [abcx-z] matches "a", "b", "c", "x", "y", or "z", as does [a-cx-z]. The - character is treated as a literal character if it is the last or the first character within the brackets: [abc-], [-abc]. The] character can be included in a bracket expression if it is the first character: []abc]. The bracket expression may also contain character classes ³ , equivalence classes ⁴ , and collating characters ⁵ .
[^]	Matches a single character that is not contained within the brackets. For example, [^abc] matches any character other than "a", "b", or "c", and [^a-z] matches any single character that is not a lowercase letter from "a" to "z". These forms can be mixed: [^abcx-z] matches any character other than "a", "b", "c", "x", "y", or "z". The - character is treated as a literal character if it is the last character or the first character after ^: [^abc-], [^-abc]. The] character is treated as a literal character if it is the first character after ^: [^]abc]. The expression may also contain character classes ⁶ , equivalence classes ⁷ , and collating characters ⁸ .
^	Matches the starting position within the string, if it is the first character of the regular expression.
\$	Matches the ending position of the string, if it is the last character of the regular expression.
*	Matches the preceding element zero or more times. For example, ab*c matches "ac", "abc", "abbbc", etc. [xyz]* matches "", "x", "y", "z", "zx", "zyx", "xyzzy", and so on.
BRE: \{m \} ERE: {m }	Matches the preceding element exactly <i>m</i> times. For example, a\{3\} matches only "aaa".
BRE: \{m , \} ERE: {m , }	Matches the preceding element at least <i>m</i> times. For example, a\{3, \} matches "aaa", "aaaa", "aaaaa", "aaaaaa", "aaaaaaa", and so on.
BRE: \{m ,n \} ERE: {m ,n }	Matches the preceding element at least <i>m</i> and not more than <i>n</i> times. For example, a\{3,5\} matches only "aaa", "aaaa", and "aaaaa". This is not found in a few older instances of regular expressions.

3 Chapter 6.0.3 on page 19

4 Chapter 6.0.5 on page 20

5 Chapter 6.2 on page 20

6 Chapter 6.0.3 on page 19

7 Chapter 6.0.5 on page 20

8 Chapter 6.2 on page 20

Metacharacter	Description
BRE: \ ERE: ()	Defines a subexpression. It is treated as a single element. For example, <code>ab*</code> matches <code>"a"</code> , <code>"ab"</code> , <code>"abb"</code> and so on, while <code>\(ab\)*</code> matches <code>""</code> , <code>"ab"</code> , <code>"abab"</code> , <code>"ababab"</code> , and so on. The string matched within the parentheses can be recalled later (see the next entry, <code>\n</code>). A subexpression is also called a marked subexpression, a block or a capturing group.
BRE only: <code>\n</code>	Matches what the n th marked subexpression matched, where n is a digit from 1 to 9. This construct is theoretically irregular (an expression with this construct does not obey the mathematical definition of regular expression), and was not adopted in the POSIX ERE syntax.

Examples:

- `.at` matches any three-character string ending with `"at"`, including `"hat"`, `"cat"`, and `"bat"`.
- `[hc]at` matches `"hat"` and `"cat"`.
- `[^b]at` matches all strings matched by `.at` except `"bat"`.
- `^[hc]at` matches `"hat"` and `"cat"`, but only at the beginning of the string or line.
- `[hc]at$` matches `"hat"` and `"cat"`, but only at the end of the string or line.
- `\[.\]` matches any single character surrounded by `"["` and `"]"` since the brackets are escaped, for example: `"[a]"` and `"[b]"`.

6.0.3 Character classes

The POSIX standard defines some classes or categories of characters as shown below. These classes are used within brackets.

POSIX class	similar to	meaning
<code>[:upper:]</code>	<code>[A-Z]</code>	uppercase letters
<code>[:lower:]</code>	<code>[a-z]</code>	lowercase letters
<code>[:alpha:]</code>	<code>[A-Za-z]</code>	upper- and lowercase letters
<code>[:digit:]</code>	<code>[0-9]</code>	digits
<code>[:xdigit:]</code>	<code>[0-9A-Fa-f]</code>	hexadecimal digits
<code>[:alnum:]</code>	<code>[A-Za-z0-9]</code>	digits, upper- and lowercase letters
<code>[:punct:]</code>		punctuation (all graphic characters except letters and digits)
<code>[:blank:]</code>	<code>[\t]</code>	space and TAB characters only
<code>[:space:]</code>	<code>[\t\n\r\f\v]</code>	blank (whitespace) characters
<code>[:cntrl:]</code>		control characters
<code>[:graph:]</code>	<code>[^[:cntrl:]]</code>	graphic characters (all characters which have graphic representation)
<code>[:print:]</code>	<code>[[:graph]]</code>	graphic characters and space

For example,

- `a[[:digit:]]b` matches `"a0b"`, `"a1b"`, ..., `"a9b"`.
- `a[:digit:]b` is an error: character classes must be in brackets
- `[[:digit:]]abc` matches any digit, `"a"`, `"b"`, and `"c"`.
- `[abc[:digit:]]` is the same as above
- `[^ABZ[:lower:]]` matches any character except lowercase letters, A, B, and Z.

6.0.4 Collating symbols

Collating symbols, like character classes, are used in brackets and have the form `[.ch.]`. Here `ch` is a digraph. Collating systems are defined by the locale.

6.0.5 Equivalence classes

Equivalence classes, like character classes and collating symbols, are used in brackets and have the form `[=a=]`. They stand for any character which is equivalent to the given. According to the standard http://pubs.opengroup.org/onlinepubs/009696899/basedefs/xbd_chap09.html#tag_09_03_03,

For example, if `'a'`, `'à'`, and `'â'` belong to the same equivalence class, then `"[=a=b]"`, `"[=â=b]"`, and `"[=à=b]"` are each equivalent to `"[aâàb]"`.

Equivalence classes, like collating symbols, are defined by the locale.

6.1 External links

- IEEE Std 1003.1, 2004 Edition. Chapter 9, Regular Expressions.⁹

6.2 Use in Tools

Tools and languages that utilize this regular expression syntax include:

- W:TextPad¹⁰

6.3 Links

- POSIX Basic Regular Expressions¹¹ at regular-expressions.info
- POSIX Basic Regular Expression Syntax¹² at boost.org

9 http://pubs.opengroup.org/onlinepubs/009696899/basedefs/xbd_chap09.html

10 <http://en.wikipedia.org/wiki/TextPad>

11 <http://www.regular-expressions.info/posix.html>

12 http://www.boost.org/doc/libs/1_44_0/libs/regex/doc/html/boost_regex/syntax/basic_syntax.html

7 POSIX-Extended Regular Expressions

The more modern "extended" regular expressions can often be used with modern Unix utilities by including the command line flag "-E".

POSIX extended regular expressions are similar in syntax to the traditional Unix regular expressions, with some exceptions. The following metacharacters are added:

Metacharacter	Description
.	Matches any single character (many applications exclude newlines, and exactly which characters are considered newlines is flavor, character encoding, and platform specific, but it is safe to assume that the line feed character is included). Within POSIX bracket expressions, the dot character matches a literal dot. For example, <code>a.c</code> matches "abc", etc., but <code>[a.c]</code> matches only "a", " ", or "c".
[]	A bracket expression. Matches a single character that is contained within the brackets. For example, <code>[abc]</code> matches "a", "b", or "c". <code>[a-z]</code> specifies a range which matches any lowercase letter from "a" to "z". These forms can be mixed: <code>[abcx-z]</code> matches "a", "b", "c", "x", "y", or "z", as does <code>[a-cx-z]</code> . The - character is treated as a literal character if it is the last or the first (after the <code>^</code>) character within the brackets: <code>[abc-]</code> , <code>[-abc]</code> . Note that backslash escapes are not allowed. The] character can be included in a bracket expression if it is the first (after the <code>^</code>) character: <code>[]abc]</code> .
[^]	Matches a single character that is not contained within the brackets. For example, <code>[^abc]</code> matches any character other than "a", "b", or "c". <code>[^a-z]</code> matches any single character that is not a lowercase letter from "a" to "z". As above, literal characters and ranges can be mixed.
^	Matches the starting position within the string. In line-based tools, it matches the starting position of any line.
\$	Matches the ending position of the string or the position just before a string-ending newline. In line-based tools, it matches the ending position of any line.
BRE: \ ERE: ()	Defines a marked subexpression. The string matched within the parentheses can be recalled later (see the next entry, <code>\n</code>). A marked subexpression is also called a block or capturing group.

Metacharacter	Description
<code>\n</code>	Matches what the n th marked subexpression matched, where n is a digit from 1 to 9. This construct is theoretically irregular and was not adopted in the POSIX ERE syntax. Some tools allow referencing more than nine capturing groups.
<code>*</code>	Matches the preceding element zero or more times. For example, <code>ab*c</code> matches <code>"ac"</code> , <code>"abc"</code> , <code>"abbc"</code> , etc. <code>[xyz]*</code> matches <code>""</code> , <code>"x"</code> , <code>"y"</code> , <code>"z"</code> , <code>"zx"</code> , <code>"zyx"</code> , <code>"xyzzy"</code> , and so on. <code>\(ab\)*</code> matches <code>""</code> , <code>"ab"</code> , <code>"abab"</code> , <code>"ababab"</code> , and so on.
BRE: <code>\{m , n \}</code> ERE: <code>{m , n }</code>	Matches the preceding element at least m and not more than n times. For example, <code>a\{3,5\}</code> matches only <code>"aaa"</code> , <code>"aaaa"</code> , and <code>"aaaaa"</code> . This is not found in a few older instances of regular expressions.

- `+` — Match the last "block" one or more times - `"ba+"` matches `"ba"`, `"baa"`, `"baaa"` and so on
- `?` — Match the last "block" zero or one times - `"ba?"` matches `"b"` or `"ba"`
- `|` — The choice (or set union) operator: match either the expression before or the expression after the operator - `"abc|def"` matches `"abc"` or `"def"`.

Also, backslashes are removed: `\{...\}` becomes `{...}` and `\(...\)` becomes `(...)`. Examples:

- `"[hc]+at"` matches with `"hat"`, `"cat"`, `"hhat"`, `"chat"`, `"hcat"`, `"ccchat"` etc.
- `"[hc]?at"` matches `"hat"`, `"cat"` and `"at"`
- `"([cC]at)|([dD]og)"` matches `"cat"`, `"Cat"`, `"dog"` and `"Dog"`

The characters `(, [, ., *, ?, +, ^, and $` are special symbols and have to be escaped with a backslash symbol in order to be treated as literal characters. For example:

`"a\.\(\(\)\)"` matches with the string `"a.)"` or `"a.("`

Modern regular expression tools allow a quantifier to be specified as non-greedy, by putting a question mark after the quantifier: `(\[\..*?\])`.

7.0.1 Character classes

The POSIX standard defines some classes or categories of characters as shown in the following table:

POSIX class	similar to	meaning
<code>[:upper:]</code>	<code>[A-Z]</code>	uppercase letters
<code>[:lower:]</code>	<code>[a-z]</code>	lowercase letters
<code>[:alpha:]</code>	<code>[A-Za-z]</code>	upper- and lowercase letters
<code>[:alnum:]</code>	<code>[A-Za-z0-9]</code>	digits, upper- and lowercase letters
<code>[:digit:]</code>	<code>[0-9]</code>	digits
<code>[:xdigit:]</code>	<code>[0-9A-Fa-f]</code>	hexadecimal digits
<code>[:punct:]</code>	<code>[.,!?:...]</code>	punctuation
<code>[:blank:]</code>	<code>[\t]</code>	space and TAB characters only
<code>[:space:]</code>	<code>[\t\n\r\f\v]</code>	blank (whitespace) characters

POSIX class	similar to	meaning
<code>[:cntrl:]</code>		control characters
<code>[:graph:]</code>	<code>[\t\n\r\f\v]</code>	printed characters
<code>[:print:]</code>	<code>[\t\n\r\f\v]</code>	printed characters and space

Links:

- [W:Regular_expression#Character_classes](#)¹
- [Character Classes that are Always Supported](#)² at boost.org

7.1 Use in Tools

Tools and languages that utilize this regular expression syntax include:

- AWK³ - uses a superset of the extended regular expression syntax

7.2 Links

- [POSIX Basic Regular Expressions](#)⁴ at regular-expressions.info
- [POSIX Extended Regular Expression Syntax](#)⁵ at boost.org

1 http://en.wikipedia.org/wiki/Regular_expression%23Character_classes
2 http://www.boost.org/doc/libs/1_44_0/libs/regex/doc/html/boost_regex/syntax/character_classes/std_char_classes.html
3 <http://en.wikibooks.org/wiki/AWK>
4 <http://www.regular-expressions.info/posix.html>
5 http://www.boost.org/doc/libs/1_44_0/libs/regex/doc/html/boost_regex/syntax/basic_extended.html

8 Non-POSIX Basic Regular Expressions

Non POSIX Basic Regular Expression Syntax : An additional non-POSIX class understood by some tools is `[:word:]`, which is usually defined as `[:alnum:]` plus underscore. This form of regular expression is used to reflect the fact that in many programming languages these characters may be used in identifiers.

Operators	
Operator	Effect
.	The dot operator matches any single character.
[]	boxes enable a single character to be matched against a character lists or character range.
[^]	A compliment box enables a single character not within in a character list or character range to be matched.
*	An asterisk specifies zero or more characters to match.
^	The caret anchor matches the beginning of the line
\$	The dollar anchor matches the end of the line

The editor vim further distinguishes *word* and *word-head* classes (using the notation `\w` and `\h`) since in many programming languages the characters that can begin an identifier are not the same as those that can occur in other positions.

(For an ASCII chart color-coded to show the POSIX classes, see ASCII¹.)

8.1 Use in Tools

Tools and languages that utilize this regular expression syntax include:

- vim²

¹ <http://billposer.org/Linguistics/Computation/ascii.html>

² <http://en.wikibooks.org/wiki/vim>

9 Emacs Regular Expressions

Notes on regular expressions used in text editor **Emacs** :

- For backslash escaping (magic vs literal), Emacs uses a mixture of BRE and ERE. Like in ERE, Emacs supports unescaped +, ?. Like in BRE, Emacs supports escaped \(\, \), \|, \{, \}.
- GNU extensions to regular expressions supported by Emacs include \w, \W, \b, \B, \<, \>, \', \ (start and end of buffer)
- No "\s" like in PCRE; whitespace is matched by "\s-".
- No "\d" like in PCRE; use [0-9] or [[:digit:]]
- No lookahead and no lookbehind like in PCRE
- Emacs regexp can match characters by syntax using mode-specific syntax tables ("\sc", "\s-", "\s ") or by categories ("\cc", "\cg").

9.1 Use in Tools

Tools and languages that utilize this regular expression syntax include:

- Emacs¹

9.2 Links

- Regular Expressions² at emacswiki.org
- Perl and Emacs regular expressions compared³ at lemoda.net

¹ <http://en.wikibooks.org/wiki/Emacs>

² <http://www.emacswiki.org/emacs/RegularExpression>

³ <http://www.lemoda.net/emacs/perl-emacs-regex/index.html>

10 Shell Regular Expressions

The Unix shell recognises a limited form of regular expressions used with filename substitution:

Operators	
Operator	Effect
?	The hook operator specifies any single character.
[]	boxes enable a single character to be matched against a character lists or character range.
[!]	A compliment box enables a single character not within in a character list or character range to be matched.
*	An asterisk specifies zero or more characters to match.

Some operators behave differently in the shell: The asterisk and hook operators do not need to follow a previous character in the shell and they exhibit non traditional regular expression behaviour.

Unsupported Constructs: Within the shell, a compliment box is formed using the pling symbol. The shell does not support the use of a caret box for character list exclusion. In the shell, a caret symbol within a box will simply be treated as one of the characters within the character list for matching.

10.1 Use in Tools

Tools and languages that utilize this regular expression syntax include:

- Bourne compatible shells

11 Implementation

11.0.1 Implementations and running times

There are at least 3 different algorithms that decide if (and how) a given string matches a regular expression. They are based on different representations of the regular expression as a Finite Automaton and on the amount of functionality present in the matcher.

1. An NFA based matcher without back-references and look ahead/behind. An input of size $O(n)$ can be tested against a regular expression of size $O(m)$ in time $O(nm)$, and additional $O(m)$ extra space by simulating an NFA using Thompson's algorithm¹. If c sub-match capture groups are to be recorded, then the running time increases to $O(nm \log c)$, but the space requirement remains $O(m)$ ².
2. An NFA based matcher with back-references and look ahead/behind. Such a matcher needs to be implemented using backtracking. An input of size $O(n)$ can be tested against a regular expression of size $O(m)$ in time $O(2^m n)$ using backtracking. Some effort is needed to ensure that the backtracking based matcher doesn't enter an infinite loop, testing the same path over and over again.
3. A DFA based matcher. DFA based matchers can't support back-references, sub-match captures, or look ahead/behind. This is the oldest and fastest kind of matcher and relies on a result in formal language theory that allows every nondeterministic Finite State Machine (NFA) to be transformed into a deterministic finite state machine (DFA). The algorithm performs or simulates this transformation and then runs the resulting DFA on the input string, one symbol at a time. The latter process (DFA matching) takes time that is proportional to the length of the input string. More precisely, a regular expression of size m on an input alphabet of size S can be converted into a DFA in time $O(2^m S)$, and subsequently an input string of size n can be tested against a DFA of any size in time $O(n)$.

The DFA based algorithm is fast to match input against a regular expression, but can be used only for matching and not for recalling grouped subexpressions. There is a variant that can recall grouped subexpressions, but its running time slows down to $O(n^2 m)$.

The running time of the backtracking based algorithm can be exponential, which simple implementations exhibit when matching against expressions like `"(a|aa)*b"` that contain both alternation and unbounded quantification and force the algorithm to consider an exponential number of subcases. More complex implementations identify and speed up various common cases where they would otherwise run slowly.

¹ <http://swtch.com/~rsc/regexp/regexp1.html>
² <https://github.com/dhruvbird/regexp-js/blob/master/complexity.md#capturing-sub-matches>

Even though backtracking implementations only give an exponential guarantee in the worst case, they allow much greater flexibility and provide more expressive power. For instance any implementation that allows the use of backreferences, or implements the various improvements that Perl introduced, must use a backtracking implementation.

Some implementations try to provide the best of both algorithms by first running a fast DFA match to see if the string matches the regular expression at all, and only in that case perform a potentially slower backtracking match.

Category:Regular Expressions³

³ <http://en.wikibooks.org/wiki/Category%3ARegular%20Expressions>

12 Glossary

This is a glossary of the book.

\A

In some flavors, the beginning of a string but not of a line in the string

\b

In some flavors, a word boundary

\B

In some flavors, a complement to `\b`

BRE

Basic regular expressions

\d

In some flavors, a digit

\D

In some flavors, a complement to `\d`

Emacs

A scriptable text editor with support for regular expressions

ERE

Extended regular expressions

GNU

A project to create a free-as-in-freedom operating system, which provides extensions to regular expressions used in tools such as Grep or Sed

Greedy

Of an operator, matching as much as it can

Grep

A command-line tool for finding lines in a text files that match a regular expression

Java

A byte-compiled programming language with support for regular expressions in its standard library since version 1.4

JavaScript

A scripting languages for the web supported by web browsers, with built-in support for regular expressions

Metacharacter

A character or sequence of characters with a special meaning, such as "." or "\+".

PCRE

Perl compatible regular expressions

Perl

An interpreted scripting language noted for its regular expressions

PHP

An interpreted scripting language with support for regular expressions

Regex

A regular expression

Regular expression

A string containing special characters indicating patterns, intended to match literal strings

\s

In some flavors, a whitespace character: space, tab, newline, form feed

\s-

In Emacs, a whitespace character

\S

In some flavors, a complement to \s

Sed

A non-interactive editor or command-line tool noted for its "s" command substituting strings that match a regular expression with other strings

\u13F

In some flavors, the character with the hexadecimal Unicode value of 13F.

Vim

A scriptable text editor with support for regular expressions

\w

In some flavors, an alphanumeric character, including "_"

\W

In some flavors, a complement to \w

\xF7

In some flavors, the character with the hexadecimal ASCII value of F7.

`\x{13F}`

In some flavors, the character with the hexadecimal Unicode value of 13F.

`\Z`

In some flavors, the end of a string but not of a line in the string

`\<`

In some flavors, an empty string before the beginning of a word

`\>`

In some flavors, an empty string after the end of a word

`^`

The beginning of a line

`$`

The end of a line

`.`

Any single character, but possibly not a newline

`[`

The opening of a character class

`]`

The closing of a character class

`(`

In some flavors, the opening of a group

`)`

In some flavors, the closing of a group

`\(`

In some flavors, the opening of a group

`\)`

In some flavors, the closing of a group

`{`

In some flavors, the opening of a match-count iterator

`}`

In some flavors, the closing of a match-count iterator

`\{`

In some flavors, the opening of a match-count iterator

`\}`

In some flavors, the closing of a match-count iterator

`|`

In some flavors, a marking of an alternative

`\|`

In some flavors, a marking of an alternative

`\1`

In some flavors, a backreference to the 1st group

`\2`

In some flavors, a backreference to the 2nd group

`*`

Any number of the previous

`+`

In some flavors, one or more of the previous

`\+`

In some flavors, one or more of the previous

`?`

In some flavors, one or none of the previous

`\?`

In some flavors, one or none of the previous

`*?`

In some flavors, a non-greedy version of `*`

`+?`

In some flavors, a non-greedy version of `+`

`}?`

In some flavors, a non-greedy version of `}`

13 Links

13.1 See also

- JavaScript/Regular Expressions¹
- Perl Programming/Regular Expressions Reference²
- Regular Expressions using R³

13.2 External Links

- regular-expressions.info⁴ - a regex tutorial site
- Text and Data Manipulation with Regular Expressions in .NET Development⁵ — tutorial, reference and *Ready to use Regular Expression patterns for VB.NET, C#.NET and ASP.NET development*
- Emacs regular expressions⁶ at cs.utah.edu
- W:Regular expression⁷
- W:Comparison of regular expression engines⁸
- W:List of regular expression software⁹

Category:Regular Expressions¹⁰

Category:Regular Expressions¹¹

1 <http://en.wikibooks.org/wiki/JavaScript%2FRegular%20Expressions>
2 <http://en.wikibooks.org/wiki/Perl%20Programming%2FRegular%20Expressions%20Reference>
3 http://en.wikibooks.org/wiki/R_Programming%2FText_Processing%23Regular_Expressions
4 <http://www.regular-expressions.info/>
5 <http://www.tipsntracks.com/182/text-and-data-manipulation-with-regular-expressions-in-net-development.html>
6 http://www.cs.utah.edu/dept/old/texinfo/emacs18/emacs_17.html
7 <http://en.wikipedia.org/wiki/Regular%20expression>
8 <http://en.wikipedia.org/wiki/Comparison%20of%20regular%20expression%20engines>
9 <http://en.wikipedia.org/wiki/List%20of%20regular%20expression%20software>
10 <http://en.wikibooks.org/wiki/Category%3ARegular%20Expressions>
11 <http://en.wikibooks.org/wiki/Category%3ARegular%20Expressions>

14 Contributors

Edits	User
3	Adrignola ¹
1	Albmont ²
85	Dan Polansky ³
1	Derbeth ⁴
1	Dirk Hünninger ⁵
9	Franl ⁶
1	Jguk ⁷
3	LoStrangolatore ⁸
1	M0rpheu\$ ⁹
66	Markhobley ¹⁰
1	Novwik ¹¹
1	PAC2 ¹²
2	Panic2k4 ¹³
1	Recent Runes ¹⁴
6	Sae1962 ¹⁵
1	Van der Hoorn ¹⁶

1 <http://en.wikibooks.org/wiki/User:Adrignola>
2 <http://en.wikibooks.org/wiki/User:Albmont>
3 http://en.wikibooks.org/wiki/User:Dan_Polansky
4 <http://en.wikibooks.org/wiki/User:Derbeth>
5 http://en.wikibooks.org/wiki/User:Dirk_H%25C3%25BCnniger
6 <http://en.wikibooks.org/wiki/User:Franl>
7 <http://en.wikibooks.org/wiki/User:Jguk>
8 <http://en.wikibooks.org/wiki/User:LoStrangolatore>
9 [http://en.wikibooks.org/wiki/User:M0rpheu\protect\char"0024\relax{}](http://en.wikibooks.org/wiki/User:M0rpheu%20protect%20char%20%220024%22relax%20%7B%7D)
10 <http://en.wikibooks.org/wiki/User:Markhobley>
11 <http://en.wikibooks.org/wiki/User:Novwik>
12 <http://en.wikibooks.org/wiki/User:PAC2>
13 <http://en.wikibooks.org/wiki/User:Panic2k4>
14 http://en.wikibooks.org/wiki/User:Recent_Runes
15 <http://en.wikibooks.org/wiki/User:Sae1962>
16 http://en.wikibooks.org/wiki/User:Van_der_Hoorn

List of Figures

- GFDL: Gnu Free Documentation License. <http://www.gnu.org/licenses/fdl.html>
- cc-by-sa-3.0: Creative Commons Attribution ShareAlike 3.0 License. <http://creativecommons.org/licenses/by-sa/3.0/>
- cc-by-sa-2.5: Creative Commons Attribution ShareAlike 2.5 License. <http://creativecommons.org/licenses/by-sa/2.5/>
- cc-by-sa-2.0: Creative Commons Attribution ShareAlike 2.0 License. <http://creativecommons.org/licenses/by-sa/2.0/>
- cc-by-sa-1.0: Creative Commons Attribution ShareAlike 1.0 License. <http://creativecommons.org/licenses/by-sa/1.0/>
- cc-by-2.0: Creative Commons Attribution 2.0 License. <http://creativecommons.org/licenses/by/2.0/>
- cc-by-2.0: Creative Commons Attribution 2.0 License. <http://creativecommons.org/licenses/by/2.0/deed.en>
- cc-by-2.5: Creative Commons Attribution 2.5 License. <http://creativecommons.org/licenses/by/2.5/deed.en>
- cc-by-3.0: Creative Commons Attribution 3.0 License. <http://creativecommons.org/licenses/by/3.0/deed.en>
- GPL: GNU General Public License. <http://www.gnu.org/licenses/gpl-2.0.txt>
- LGPL: GNU Lesser General Public License. <http://www.gnu.org/licenses/lgpl.html>
- PD: This image is in the public domain.
- ATTR: The copyright holder of this file allows anyone to use it for any purpose, provided that the copyright holder is properly attributed. Redistribution, derivative work, commercial use, and all other use is permitted.
- EURO: This is the common (reverse) face of a euro coin. The copyright on the design of the common face of the euro coins belongs to the European Commission. Authorised is reproduction in a format without relief (drawings, paintings, films) provided they are not detrimental to the image of the euro.
- LFK: Lizenz Freie Kunst. <http://artlibre.org/licence/lal/de>
- CFR: Copyright free use.

- EPL: Eclipse Public License. <http://www.eclipse.org/org/documents/epl-v10.php>

Copies of the GPL, the LGPL as well as a GFDL are included in chapter Licenses¹⁷. Please note that images in the public domain do not require attribution. You may click on the image numbers in the following table to open the webpage of the images in your webbrowser.

¹⁷ Chapter 15 on page 45

15 Licenses

15.1 GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

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

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure that you remain free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

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.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS 0. Definitions.

"This License" refers to version 3 of the GNU 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. 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. 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. 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. 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. 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 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. 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 to use, 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 specify 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. 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. 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. 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. 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. 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 promise 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. 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. 13. Use with the GNU Affero General Public License.

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 Affero 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 special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such. 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU 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 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 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 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.

15.2 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright (c) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference. 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses

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. 15. Disclaimer of Warranty.

THESE ARE NO WARRANTIES 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. 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. 17. Interpretation of Sections 15 and 16.

following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History"). To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License. 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies. 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first one listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general networking-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of additional material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

If it is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document. 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version precisely as the full text of the Invariant Sections and required Cover Texts given in the Document's license notice. In addition, you must do these things in the Modified Version:

- * A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- * B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- * C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- * D. Preserve all the copyright notices of the Document.
- * E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- * F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the medium below.
- * G. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions if they were based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- * K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- * L. Preserve all the Invariant Sections of the Document, unaltered in their text and

if the disclaimer of warranty and limitation of liability provided above cannot be made legal local 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 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 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 General Public License for more details.

in their titles. Section numbers or the equivalent are not considered part of the section titles. * M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version. * N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section. * O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or of the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added (by or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version. 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements". 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document. 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate. 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title

You should have received a copy of the GNU 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 the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

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 GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/why-not-lgpl.html>>.

(section 1) will typically require changing the actual title. 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

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 (or any time) 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, receipt of a copy of some or all of the same material does not give you any rights to use it. 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See <<http://www.gnu.org/copyleft/>>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document. 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public webkit that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

15.3 GNU Lesser General Public License

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

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

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

0. Additional Definitions.

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

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

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

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

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

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

1. Exception to Section 3 of the GNU GPL.

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

2. Conveying Modified Versions.

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

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

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

3. Object Code Incorporating Material from Library Header Files.

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

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

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

4. Combined Works.

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

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

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

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

* d) Do one of the following:

- o 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
- o 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
- * e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

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

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

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

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

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

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

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