

# Practical Regular Expressions

Version 1.1

Copyright 2002 Brenda J. Butler

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License , Version 1.1 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 this license is included in the section entitled "GNU Free Documentation License".

# Practical Regular Expressions

- Brenda Butler
- bjb at achilles dot net
- Unix programmer since about 1990
- used Linux since about 1998
- I'm not a guru on:
  - regular expressions
  - awk
  - sed
  - perl
- But I can use them to do some stuff and that's what I'm going to show you

# Practical Regular Expressions: Overview

- What are Regular Expressions for?
- What tools use Regular Expressions?
- How to write Regular Expressions
- About the tools that use Regular Expressions
- Examples and practice

# Practical Regular Expressions

- Kinds of Regular Expressions
- Basic Regular Expressions
- Extended Regular Expressions
- Historical differences
- POSIX standards
- GNU: same functionality between Basic and Extended R.E.s (but different syntax)
- We will look at Regular Expressions from a GNU point of view

# What are Regular Expressions for?

- To search through or edit text files
- Looking for strings that are not always exactly the same, but can be described in terms of a pattern
  - E.g., telephone numbers: 3 digits, a hyphen, 4 digits
  - 999-9999 is a regular expression of sorts.
  - So is yy/mm/dd, or yyyy/mm/dd, or yyyy/mmm/dd
  - IP addresses: xx.xx.xx.xx
- Regular expressions are used to find matching strings in an existing text, not to generate all strings that match the expression.

# What tools use Regular Expressions?

- Quite a few tools can use Regular Expressions
  - grep - look through text file for lines that match a pattern
  - sed - stream editor
  - awk - text processing programming language
  - perl - programming language specialised for text processing among other things
  - Various editors: emacs, vim, nedit, ...
  - And more...

# Regular Expression

## atoms

Most basic Regular Expression: exact match.

- cheesecake

WARNING: Regular Expressions are case-sensitive

Match on character classes:

- `chee[sp]eca[rk]e`

would match

- cheepecake
- cheepicare
- cheesecake
- cheesecare

# Regular Expression

## atoms

Match on the negation of a character class:

- `From[ ^ : ]`

would match any string containing From that isn't followed by colon.

It would match

`From :`

but not

`From:`

This only works within the square brackets.



# Regular Expression

## atoms

Match on character classes (with range):

- `chee[a-z]ecake`

matches:

- `chee a cake`
- `chee b cake`
- `chee .. cake`
- `chee s cake`
- `chee .. cake`
- `chee y cake`
- `chee z cake`

# Regular Expression

## atoms

Match on character classes (with multiple ranges):

- `chee[a-zA-Z]ecake`

matches:

- `chee aecake`
- `chee becake`
- `chee ..ecake`
- `chee zecake`
- `chee Aecake`
- `chee Becake`
- `chee ..ecake`
- `chee Zecake`

# Regular Expression

## atoms

To include ], [ or - in a character class, put the - at the beginning or end of the character class. Put the ] at the beginning of the character class. Put the ^ anywhere but first in the character class.

- `[]` To match ]
- `[]-]` To match ] or -
- `[A-Za-z0-9_ -]` to match alphanumeric, underscore or hyphen
- `[$ ^]` to match \$ or ^

# Regular Expression

## atoms

WARNING! The range is based on the character encoding. Since the ASCII lower-case a-z are coded as sequential numbers, and the ASCII upper-case A-Z are coded as sequential numbers, and the ASCII numerals are coded as sequential numbers, this works. But be careful when using ranges such

as [0-f] (hexadecimal digits). This does not work. What you need to say is [0-9a-f].

0 = 48   A = 65   a = 97

1 = 49   B = 66   b = 98

...   ...   ...

9 = 57   Z = 90   z = 122

(See the `ascii` man page:)

(`man ascii`)

# Regular Expression

## atoms

Match on any character:

- `chee . ecake`

matches:

- `chee a ecake`
- `chee b ecake`
- `chee .. ecake`
- `chee A ecake`
- `chee B ecake`
- `chee .. ecake`
- `chee 0 ecake`
- `chee [ ecake`
- etc

# Regular Expression

## grouping

You can put a group of atoms together to make a unit, with parentheses:

- (cherry)cheesecake

This matches

cherrycheesecake

- -- the same as an exact match. But with the group, you can do more stuff, like repetition and alternation.

# Regular Expression modifiers

Match possible repetitions:

baa\*

matches

ba

baa

baaa

baaaa

BEWARE: a\* matches 0 or more a's

BEWARE: \* by itself does not match anything (not like in the shell)

# Regular Expression modifiers

Match possible repetitions:

```
Package: .*ftp
```

matches lines which contain the word "Package:" followed by any string followed by the string "ftp", for instance:

```
Package: tftp
```

```
Package: ftp
```

```
Package: ftp-gtk
```

BEWARE: .\* matches **0** or more characters

BEWARE: \* by itself doesn't match anything (unlike the shell)

BEWARE: \* is greedy



# Regular Expression modifiers

Writing

```
[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]
```

is verbose and error-prone. We can write instead:

```
[0-9]{3}-[0-9]{4}
```

IP address could be:

```
([0-9]{1,3}\.){3,3}[0-9]{1,3}
```

There! Isn't that easier to read? :-)

This allows you to say that the ip numbers can be 1 to 3 digits long.

# Regular Expression modifiers

Two types of limited repetitions are "one or more occurrence" and "zero or one occurrence". They have their own shorthand:

```
(cherry)?cheesecake
```

matches cheesecake and cherrycheesecake.

```
(very)+cherrycheesecake
```

matches verycherrycheesecake,

veryverycherrycheesecake,...

# Regular Expression modifiers

Match possible repetitions of syllables:

- `ba(na)*`

matches

ba

bana

banana

bananana

...

BEWARE: `(na)*` matches **0** or more na's

# Regular Expression modifiers

Match different alternatives: "Alternation":

`ba | i`

matches

`ba`

`i`

NOTE the `|` modifier applies to all the adjacent atoms of a group

# Regular Expression modifiers

Match different alternative syllables: "Alternation with grouping"

```
(cherry|chocolate)cheesecake
```

matches

```
cherrycheesecake
```

```
chocolatecheesecake
```

Note the | modifier applies to all the atoms in the adjacent group.

```
cherry|chocolate cheesecake
```

matches

```
cherry
```

```
chocolate cheesecake
```

# Regular Expression

## anchors

Match only at the beginning or end of a string:

`^Package:`

matches

Package: xserver-common

Package: wu-ftp

Package: gnome-games

... all the lines that begin with Package:

`d$`

matches

Package: wu-ftp

Package: bind

... all the lines that end with d

# Regular Expression

## anchors

BEWARE! `^` and `$` are only anchors when they appear at the beginning or end of the pattern, respectively. Otherwise, they are an atom that matches themselves.

`^$`

matches a blank line

`$^`

matches any line with `$^` within it. Actually you still have to escape the special characters `$` and `^`:

`\$ \^`

# Regular Expression anchors

You can anchor to the beginning or end of a word with

`\<` and

`\>`

That makes us wonder, what is a "word"?

"Word-constituent characters are letters, digits, and the underscore" according to the man page for grep.

`\<the\>`

will find occurrences of the whole word "the", and will not find occurrences of "the" buried in other words like "other". It will find the at the beginning or end of lines.



# Regular Expression backreferences

You can match repeating substrings with backreferences. A back reference looks like this:

```
(n...)\1
```

The \1 part is the back-reference. This matches all 8-letter sequences starting with n, whose second four letters are the same as the first four letters. So it would match

- nanonano
- noonnoon

but not

- nanonoon

The backreference is numbered, refers to nth left parenthesis.

There can be 9 of them (single digit only).

# Examples and Practice

Match a phone number:

```
[0-9][0-9][0-0]-[0-9][0-9][0-9][0-9]  
[0-9]{3}-[0-9]{4}
```

Match a date: (YY/MM/DD format)

```
([0-9]{2}/){2}[0-9]{2}
```

Match an ip address:

```
[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}
```

Note that the . has been escaped, otherwise it would be interpreted as a wildcard.

# grep

grep looks for a pattern on each line of a file.

It prints each line of a file that contains the pattern.

With command line options, you can make it do slightly different things, like print the line numbers of lines that match, or print the contents or line number of lines that don't match.

You can also use the -E or -F options to have grep interpret the pattern as an extended regular expression or fixed string.

# Examples with grep

Looking for all mouse drivers:

in /dev:

```
$ ls -la /dev/psaux
```

shows

```
crw----- root 10, 1  Mar 17 11:42 /dev/psaux
```

We see that 10 is the major number for the mouse device. We can search through the /dev directory for all devices with major number 10 to find all the mouse devices:

```
$ ls -la /dev | grep 10,
```

```
crw----- root 10, 0  Aug 30 2001 logibm
crw----- root 10, 0  Aug 30 2001 logimouse
crw----- root 10, 1  Mar 17 11:42 psaux
crw----- root 10, 1  Aug 30 2001 psmouse
crw----- root 10, 2  Aug 30 2001 msmouse
crw----- root 10, 2  Aug 30 2001 inportbm
crw----- root 110, 0  Aug 30 2001 srnd0
```

# Examples with grep

But we see we found an extra item, srnd0 which matched our search string

```
10,
```

Refine the search string: " 10," (put a space at the front).

```
$ ls -la /dev | grep " 10,"
crw----- root  10, 0  Aug 30  2001 logibm
crw----- root  10, 0  Aug 30  2001 logimouse
crw----- root  10, 1  Mar 17 11:42 psaux
crw----- root  10, 1  Aug 30  2001 psmouse
crw----- root  10, 2  Aug 30  2001 msmouse
crw----- root  10, 2  Aug 30  2001 inportbm
```

We found only the mouse devices this time.

# Examples with grep

The file `/var/log/messages` is one place where the system logs error, debug, and informational messages. It can grow quite large. One of the jobs of the system administrator is to keep an eye on it to catch any unauthorised activity. It can be useful to grep through it, and display only those lines that you aren't expecting.

Look in the file `messages` in your home directory, and create some regular expressions to match some of the lines. Then use `grep` or `egrep` to display all the lines in `messages` except those lines. Check the man page to find out what the option is to show all lines except the ones that match.

# sed

- How sed works and what it's for.
  - Call sed with a script and a text file as input
  - Sed follows the instructions in the script and applies them to the text file, one line at a time (one "cycle").
  - Sed uses a "pattern space" and a "hold space". Most operations happen in the pattern space.
  - You can have sed print the pattern space after each cycle, or you can have sed print out only when you tell it to print.
  - There are commands for moving data between the pattern space and the hold space.
  - You can accumulate more than one line in the pattern or hold spaces, separated by newlines.

# Examples with sed

- Convert from unix linefeeds to DOS linefeeds.
- File contains:

Joe Clark 555-4382

Pierre Trudeau 555-8921

Jean Chretien 555-5512

Brian Mulroney 555-2398

- Want to have

Joe Clark 555-4382 ^M

Pierre Trudeau 555-8921 ^M

Jean Chretien 555-5512 ^M

Brian Mulroney 555-2398 ^M

```
sed 's/$/\015/' phones.txt
```



# Examples with sed

- Convert from unix linefeeds to DOS linefeeds.

```
sed 's/$/\015/' phones.txt
```

- But sed does not understand \015 as a substitute for "carriage return".
- I'm not sure how to type ^M at the command line, so I have to create a sed script u2d.sed that looks like this

```
s/$/^M/;
```

- and run it like this:

```
sed -f u2d.sed phones.txt
```

- To see the results

```
sed -f u2d.sed phones.txt | od -c
```

# Examples with sed

- Go through a text file which has phone numbers without area codes, and put in area codes.
- File contains:

```
Joe Clark 555-4382  
Pierre Trudeau 555-8921  
Jean Chretien 555-5512  
Brian Mulroney 555-2398
```

- Want to have

```
Joe Clark (613) 555-4382  
Pierre Trudeau (613) 555-8921  
Jean Chretien (613) 555-5512  
Brian Mulroney (613) 555-2398
```

# Examples with sed

```
sed " s/ [0-9][0-9][0-9]-[0-9][0-9][0-9][0-9] / (613) & /" phones.txt
```

Note that the stuff in the replacement string is not a regular expression. It has its own syntax.

Notice also that we used the verbose way of writing the Regular Expression. Due to differences in the way different programs write Regular Expressions, we would have to write this to use the limits method:

```
sed "s/\([0-9]\)\{3\}-\([0-9]\)\{4\}/(613) &/" phones.txt
```

Try it both ways. Use the man page of the tool you're using to find out exactly how to write the Regular Expression.

# awk

- How awk works and what it's for.
  - Call awk with a script and a text file as input.
  - Awk looks at each line in the text file one after the other.
  - It uses the script to tell it what to do with each line.
  - The script consists of a series of "patterns" and "actions".
  - When a line matches the pattern, awk performs the action.
  - There is an extra clause at the beginning called BEGIN and one at the end called END for initialisation and cleanup actions.
  - Awk can do a few more things than sed: for instance it can do arithmetic. The actions can be just about anything you can do in a programming language, except moving around in the input text file.

# awk

Built-in variables:

RS	Input record separator
FS	Input field separator
\$0, \$1, \$2...	Input record, fields
NF	Number of input fields
ORS	Output record separator
OFS	Output field separator
RT	Record terminator

and lots more...

# Examples with awk

- In DOS or Windows, lines in text files are ended with "\r\n". In unix, lines are ended with "\n". Here's an awk script to convert from unix to dos.

```
{  
    print $0 "\015"  
}
```

# Examples with awk

Counting the lines in phones.txt.

```
awk "BEGIN {count = 0;} {++count;}  
END {print count;}" phones.txt
```

will print out

4

# Examples with awk

Looking for all occurrences of string imageN in a file and replacing it with image(N+1).

The awk script is:

```
{
  if ( match ($0, "image[0-9]+") ) {
    $id = substr ($0, RSTART+5, RLENGTH-5);
    $id = $id + 1;
    $idstring = "image" $id;
    sub ("image[0-9]+", $idstring);
  }
  print;
}
```

This needs some work.



# Examples with awk

```
BEGIN {
    FS = image
}
{
    for ( ii = 2; ii <= NF; ++ii ) {
        if ( match ($ii, "[0-9]+") ) {
            num = substr($ii,RSTART,RLENGTH)
            ++num;
            sub ("^[0-9]+", num, $ii);
        }
    }
    print
}
```

# perl

- How perl works and what it's for.
  - Call perl with a script as input. You can also give it other arguments (such as input files).
  - Perl does what you tell it to in the script. There is a mode where it can be told to read each input file one line at a time, and print it or not (as in sed or awk).
- Perl can do anything sed or awk can do, and more.

When using perl, always specify `-w` and always `"use strict;"`

```
#!/usr/bin/perl -w
```

```
use strict;
```

# Examples with perl

The awk example in perl:

Looking for all occurrences of string imageN in a file and replacing it with image(N+1).

```
#!/usr/bin/perl -w
use strict;
use vars qw( $id );
while (<>) {
    if ( /image([0-9]+)\.jpg/ ) {
        $id=$1;
        ++$id;
        s/image([0-9]+)\.jpg/image$id.jpg/;
    }
    print;
}
```

This doesn't catch the second image on the line at the end of the file!

# Examples with perl

Printing out all the lines of a file in reverse order.

```
#!/usr/bin/perl -w
use strict;
use vars qw( @lines );

while (<>) {
    unshift @lines, $_;
}
print @lines;
```

# emacs

Search through a file for anything matching a regular expression:

```
M-x search-forward-regexp [return] regular expression  
[return]
```

Go through the file searching for matches to a regular expression, and ask if it should be replaced with a string:

```
M-x query-replace-regexp [return] regular expression  
[return] replacement string [return]
```

# vim

Search through a file for anything matching a regular expression:

```
/regular expression[return]
```

Go through the file searching for matches to a regular expression, and ask if it should be replaced with a string:

```
:s/regular expression/replacement string/
```

# regcomp (in C)

- There are C library functions for compiling regular expressions and applying them to text in your program. See `regex(3)`.
- SYNOPSIS

```
#include <sys/types.h>
```

```
#include <regex.h>
```

```
int regcomp(regex_t *preg, const char *regex, int cflags);
```

```
int regexec(const regex_t *preg, const char *string,  
            size_t nmatch, regmatch_t pmatch[], int  
            eflags);
```

```
size_t regerror(int errcode, const regex_t *preg, char  
               *errbuf, size_t errbuf_size);
```

```
void regfree(regex_t *preg);
```

# Basic vs Extended Regular Expressions

- There are Basic and Extended Regular Expressions
- Different docs give different syntaxes for each. Some docs say that BRE have a bunch of extra backslashes while other docs say that the ERE have the extra backslashes.
- one might look like:
  - `\(Fri\|Tue\)`
  - `[0-9]\{3\}`
- the others look like:
  - `(Fri|Tue)`
  - `[0-9]{3}`
- It all depends on the individual program! READ THE MAN PAGE for each tool (and each OS)! See the gawk and grep info pages.



# Resources

- OCLUG mailing list: [oclug@lists.oclug.on.ca](mailto:oclug@lists.oclug.on.ca)
- web pages
- man pages
- books
- the scripts in `/etc` and `/etc/init.d`

# **Are you a touch typist?**

Now's your chance to perfect the punctuation  
keys

# Where the keys are

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /

# Regular Expression

## Special Characters

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /

~ `

~	!	@	#	\$	%	^	&	*	(	)	_	+		BS
`	1	2	3	4	5	6	7	8	9	0	-	=	\	
	Q	W	E	R	T	Y	U	I	O	P	{	}		
											[	]		
	A	S	D	F	G	H	J	K	L	:	"			
										;	'			
	Z	X	C	V	B	N	M	<	>	?				
								,	.	/				

**! 1**

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /

@ 2

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /

# # 3

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /



# \$ 4

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /

# % 5

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /

^ 6

~	!	@	#	\$	%	^	&	*	(	)	_	+		BS
`	1	2	3	4	5	6	7	8	9	0	-	=	\	
	Q	W	E	R	T	Y	U	I	O	P	{	}		
											[	]		
	A	S	D	F	G	H	J	K	L	:	"			
										;	'			
	Z	X	C	V	B	N	M	<	>	?				
								,	.	/				

# & 7

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /

# \* 8

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /

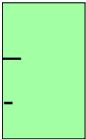

( 9

~	!	@	#	\$	%	^	&	*	(	)	-	+		BS
`	1	2	3	4	5	6	7	8	9	0	-	=	\	
	Q	W	E	R	T	Y	U	I	O	P	{	}		
											[	]		
	A	S	D	F	G	H	J	K	L	:	"			
										;	'			
	Z	X	C	V	B	N	M	<	>	?				
								,	.	/				

) 0

~	!	@	#	\$	%	^	&	*	(	)		+		BS
`	1	2	3	4	5	6	7	8	9	0		=	\	
	Q	W	E	R	T	Y	U	I	O	P	{	}		
											[	]		
	A	S	D	F	G	H	J	K	L	:	"			
										;	'			
	Z	X	C	V	B	N	M	<	>	?				
								,	.	/				

— —

~ ! @ # \$ % ^ & \* ( )  + | BS  
` 1 2 3 4 5 6 7 8 9 0  = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /



+ =

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /

| \

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /

{ [

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /


} ]

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /

⋮ ;

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : ;  
Z X C V B N M < > ?  
, . /

!! !

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "   
; '   
Z X C V B N M < > ?  
, . /

< ,

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
. /

> .

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /



? /

~ ! @ # \$ % ^ & \* ( ) \_ + | BS  
` 1 2 3 4 5 6 7 8 9 0 - = \  
Q W E R T Y U I O P { }  
[ ]  
A S D F G H J K L : "  
; '  
Z X C V B N M < > ?  
, . /

# GNU Free Documentation License

GNU Free Documentation License  
Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA 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 written 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 that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

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. (For example, 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.

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 "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and 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 has been designed to thwart or discourage subsequent modification by readers is not Transparent. 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 designed for human modification. Opaque formats include PostScript, PDF, 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 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.

## 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 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 ones 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 publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the

general network-using public has access to download anonymously at no charge using public-standard network protocols. 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.

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 under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. 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 less than five).
- 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 Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. 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 it was 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. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, 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 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 as "Endorsements" or to conflict in title with any Invariant Section.

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 that 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.

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, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they 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 quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around 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 provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 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.

## 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 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts. 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.