

Regular Expression Contents

Regular Expression

Regular Expression Patterns

Regular Expression Verbs

Regular Expression Utilities

Regular Expression Demo

Regular Expression Copyright

Regular Expression

A regular expression is a text string that specifies a pattern of characters. This facility allows you to write J programs using regular expressions to search arbitrary text. You can perform a search for a single or multiple matches, or extract the matched text from the string. You can also merge new text in to replace the matches, or apply a verb to the matched text in a string.

Utility verbs are included to assist in building regular expression patterns, as is a utility to search J scripts and other text files for patterns.

The primary definitions are in `system\main\regex.ijs` ; utility verbs to build patterns are in `system\packages\regex\regbuild.ijs`.

The Find in Files utility, available from the Edit menu or by pressing Ctrl+Shift+F, can search for simple text, regular expression, or special patterns such as the assignment of a name.

Labs and a demo on regular expressions are available from the **Studio** menu.

Patterns

Verbs

Utilities

Demo

Copyright

Regular Expression Patterns

A regular expression pattern is a sequence of elements which matches successive portions of a character string. For example, simple letters are elements which match the same characters in the string. The asterisk indicates that the previous element should be matched 0 or more times. So, a pattern of `abcd` must match in the string exactly; a pattern of `ab*cd` matches the letter `a` followed by 0 or more occurrences of the letter `b`, followed by the letters `cd`. The particular elements of a pattern are described below.

Characters

Non-special characters match exactly. Non-special characters are anything other than:

`[] () { } $ ^ . * + ? | \`

A special character is included as simple text by preceding it with a backslash.

Character sets

The special character `.` matches any character (except the null character, `0{a.}`)

The special characters `^` and `$` match the start and end of lines.

Sets of characters are defined by enclosing the list of characters in brackets:

`[aeiou]` matches a single vowel character

Ranges can also be included within the brackets:

`[a-z]` matches any lower case letter

Combinations of the above are acceptable:

`[a-zA-Z13579]` matches any lower case, upper case, or odd digit

Fixed sets (classes) of characters can be included in the list, as a name within bracket-colon pairs:

`[#:digit:]abc` matches the character `#`, a digit, or any of the letters `a`, `b`, or `c`

The character classes defined are:

<code>alnum</code>	alphanumeric	<code>alpha</code>	alphabetic
<code>blank</code>	tab and space	<code>cntrl</code>	control chars
<code>digit</code>	digits	<code>graph</code>	printable (except space)
<code>lower</code>	lowercase	<code>print</code>	printable
<code>punct</code>	punctuation	<code>space</code>	whitespace
<code>upper</code>	uppercase	<code>xdigit</code>	hex digits

If a set begins with `^`, then the pattern will match with any character *not* in the set.

Subexpressions

A series of elements may be combined by enclosing them in parenthesis. Subexpressions are affected by closures such as `*` just as simple characters are:

`([a-z][0-9])*` matches any number of occurrences of a letter followed by a digit

The result of searches for a pattern return a match for the overall pattern, and a separate match for each subexpression

A `\` followed by a digit, `N`, matches the same substring which occurred in the `N`th subexpression:

`([:digit:])#\1` matches one or more digits, followed by a `#`, followed by the same string of digits

Closures

A `*` following an element matches 0 or more occurrences of that element:

`[aeiou]*` matches 0 or more vowels

A `+` following an element matches 1 or more occurrences of that element:

`[[:alpha:]]+` matches 1 or more alphabetic characters

A `?` following an element matches 0 or 1 occurrences of that element:

`-?[:digit:]]+` matches an optional hyphen, followed by 1 or more digits

An interval expression, `{m,n}`, follows an element to allow it to match at least `m`, and no more than `n`, occurrences of the element:

`[[:digit:]]{3,5}` matches 3, 4, or 5 digits

Alternation

Multiple regular expressions can be separated with a vertical bar `|` to match any of them:

`print|list|exit` matches any of the strings `print`, `list`, and `exit`

Matches

When searching for a pattern in a string, it is possible to find multiple substrings which match the pattern. The one that is returned is the one which starts earliest in the string. If more than one match starts at the same place, the longest one is returned.

Even once a particular match is located, it is possible for there to be multiple combinations of the contents of the subexpressions which make it up. As a rule, whenever possible the subexpressions which begin earlier in the string will be as long as possible.

The result of a match is a table which describes the match. The first row covers the whole match, and subsequent rows describe where the subexpressions in the pattern match in the string. Each row has two elements: index of the first character of the start of the match, and the length of the match. Any row which doesn't participate in the match is filled with `_1 0`.

Regular Expression Verbs

The standard regex verbs are defined in `system\main\regex.ijs`. The main verbs are `rxmatch` and `rxmatches`. The former locates the first occurrence of a match in the string; the latter locates all occurrences. Four other verbs create, list, display, and free up compiled patterns: `rxcomp`, `rxhandles`, `rxinfo`, and `rxfree`.

Most of the rest of the definitions either use the `rxmatch` or `rxmatches` verbs, or take the result of them as arguments.

`match = . pattern rxmatch string` Find first match

The result of `rxmatch` is a table, each row being an index/length pair. The first row describes the entire match, one row per subexpression follow which describes where each subexpression was found in the string. Where a match does not occur, `_1 0` is returned.

`matches = . pattern rxmatches string` Find all matches

`rxmatches` returns a list of tables, with one item per match in the string. The shape of the result is `#matches` by `#subexpr` by 2.

`phandle = . rxcomp pattern` Compile pattern
`rxfree phandle` Release compiled pattern
`phandles = . rxhandles ''` Return all pattern handles
`'nsub pat' = . rxinfo phandle` Return #subexprs ; pattern

The verbs `rxcomp`, `rxhandles`, `rxinfo`, and `rxfree` allow you to create pattern handles which are simple integers which represent compiled patterns. A handle can be used anywhere a pattern can be and, if used repeatedly, will avoid having to recompile the pattern on each call.

`rxcomp` compiles a pattern and returns a handle.

`rxhandles` returns a list of all existing handles.

`rxinfo` returns information about a handle. It currently returns a boxed list of `1 +` the number of subexpressions and the original pattern. The length of the result may be extended (on the right) in the future.

`rxfree` releases all resources associated with a compiled pattern.

`errtext = . rxerror ''` Error text

The result of `rxerror` is a text string describing the last error from a regular expression verb.

`ismatch = . pattern rxeq string` 1 if entire string matches

Returns a 1 if the pattern fully describes the string. (Similar to `=` verb).

`index = . pattern rxindex string` index of match or #string

The result of `rxindex` is the index of the first match, or #string if none. (Similar to `i . verb`).

`mask = . pattern rxE string` mask: 1's start matches

`rxE` returns a boolean mask of length #string, with 1's to mark the start of a match. (Similar to `E . verb`).

`sub = . pattern rxfirst string` first substring match

`rxfirst` returns the substring in the right argument which matches the pattern.

`subs = . pattern rxall string` all substring matches

The result of `rxall` is a boxed list of all substrings in the right argument which match the pattern.

`subs = . matches rxfrom string` select substrings matched

`rxfrom` returns a box containing the substrings described by each index/length pair on the left.

`subs = . matches rxcut string` cut into alternating non-match/match

`rxcut` returns a boxed list which will match the original string if razed. The items alternate between non-matches and matches, always starting with a non-match.

`newstr = . string rxrplc (pat;rplcstr)` replace pat with rplcstr

`rxrplc` replaces substrings in the left argument. The right argument is a boxed list of the pattern and the replacement text.

`newstr = . rplcstrs matches rxmerge string merge rplcstrs into string`

`rxmerge` takes a table of matches as an argument, and returns a verb which merges the boxed strings in the left argument into those positions on the right. (Similar to `} adverb`).

`newstr = . pattern f rxapply stringapply f to each match`

`rxapply` applies its verb argument to each of the substring in the right argument which match the pattern in the left argument.

All verbs which take a pattern as an argument can be called with either a character list containing a pattern or pattern handle (an integer resulting from `rxcomp`). For example,

`'[[:alpha:]]+' rxmatches str` NB. match all sets of letters in str

```

handle=. rxcomp '[:,alpha:]]+'    NB. compile pattern into handle
handle rxmatches str    NB. do the match
rxfree handle    NB. (once handle is no longer required)

```

Notes

1. the `rmatch` and `rxmatches` verbs return either a single or list of matches, respectively, with each match being a table of index/length pairs for the match and each subexpression. Other verbs which use the result of `rxmatch` or `rxmatches` tend to only use the first row for each match, which represents the entire match.

2. if you're interested in one or more of the subexpressions, it is possible to identify the specific rows of the match which are to be returned by `rxmatch` and `rxmatches`. If a boxed array is passed rather than a character or numeric pattern, it is a 2-element list consisting of a pattern and a list of the indices of the important rows in a match.

For example, the pattern `'(x+)([:,digit:]]+)'` matches one or more letters 'x', followed by a string of digits, with both the 'x's and the digits being a subexpressions of the pattern. Each match will be returned as a three-row table, describing the entire match, just the 'x's, and just the digits.

```

pat=. rxcomp '(x+)([:,digit:]]+)'
str=. 'just one xxx1234 match here'
pat rxmatches str
9 7
9 3
12 4
(pat;1 2) rxmatches str    NB. just the 'x's and digits
9 3
12 4

```

```

pat |. rxapply str    NB. reverse the whole match
just one 4321xxx match here
(pat;2) |. rxapply str    NB. reverse just the digits
just one xxx4321 match here

```

Examples

```

pat=. '[:,alpha:]][:,alnum:]_)*'    NB. pattern for J name
str=. '3,foo3=.23,j42=.123,123'    NB. a sample string
pat rxmatch str    NB. find at index 2, length 4
2 4

```

```

pat=. '[:,alpha:]][:,alnum:]_)* *=[.;]'
```

NB. subexp is name in assign

```

pat rxmatch str    NB. pattern at 2/6; name at 2/4
2 6
2 4

```

```

pat rxmatches str    NB. find all matches

```

```

2 6
2 4

11 5
11 3

    ]phandle=. rxcomp pat    NB. compile
1
    rxcomp '[wrong'    NB. a bad pattern
|domain error: rxcomp
|    rxcomp'[wrong'

    rxerror ''
Unmatched [ or [^

    rxhandles ''    NB. just handle 1 defined
1

    rxinfo phandle    NB. return (1+#subexp);pattern
+-----+
|2|([[[:alpha:]]|[:alnum:]_]*)*=[.;]|
+-----+

    phandle rxmatches str    NB. use phandle like pattern
2 6
2 4

11 5
11 3

    phandle rxfirst str    NB. first matching substring
foo3=.

    phandle rxall str    NB. all matching substrings
+-----+-----+
|foo3=.|j42=.|
+-----+-----+

    phandle rxindex&> ' foo=.10';'nothing at all'    NB. index of match
2 14

    phandle rxE str    NB. mask over matches
0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0

    '[[[:digit:]]*' rxeq '2342342'    NB. test for exact match
1
    '[[[:digit:]]*' rxeq '2342 342'
0

```

```
    phandle rxmatch str
2 6
2 4
```

Entire and subexpression match

```
    m=. phandle rxmatches str
    phandle rxmerge str
+-----+-----+
|foo3=.|foo3|
+-----+-----+
```

```
    phandle rxmatches str    NB. all matches
2 6
2 4
```

```
11 5
11 3
```

```
    ]m=. (phandle; ,0) rxmatches str    NB. entire matches only
2 6
```

```
11 5
```

```
    m rxcut str    NB. return alternating non-match/match boxes
```

```
+--+-----+--+-----+-----+
|3,|foo3=.|23,|j42=.|123,123|
+--+-----+--+-----+-----+
```

```
    phandle |. rxapply str    NB. reverse each match
3, .=3oof23, .=24j123,123
```

```
    (phandle; ,1) |. rxapply str    NB. reverse just name part of match
3,3oof=.23,24j=.123,123
```

Regular Expression Utilities

The script `system\packages\regex\regbuild.ijs` contains definitions to for building regular expression patterns.

Many of the verbs below may enclose its argument in parentheses (to make it a subexpression). For example,

```
anyof 'abc' returns '(abc)*'.
```

The argument is only put in parentheses if necessary.

```
anyof set 'abc' is '[abc]*'.
```

The following verbs correspond directly to a feature of the regular expression notation:

```
set chars          returns set construction for chars
  set 'abc'
[abc]
```

```
not chars          set of non-matching chars
  set not 'abc'
[^abc]
```

```
sub pat            make a subexpression
  set 'abc'
(abc)
```

```
someof pat         pattern matching 1 or more pat
  someof 'abc'
(abc)+
```

```
anyof pat          pattern matching 0 or more pat
  anyof 'abc'
(abc)*
```

```
optional pat       pattern matching 0 or 1 pat
  optional 'abc'
(abc)?
```

```
(min,max) of pat  pattern matchin min up to max of pat
  2 4 of 'abc'
(abc){2,4}
```

```
pat1 or pat2       pattern matching either pat1 or pat2
  'abc' or 'd'
abc|x
```

```
pat1 by pat2       pattern matching pat1 immediately followed by pat2
  'action=' by 'move' or 'copy'
```

```
action=(move|copy)
```

```
sub pat          makes pat a subexpression
```

```
  sub 'abc'
```

```
(abc)
```

```
bkref refnum    back-reference to a previous subexpression
```

```
  bkref 1
```

```
\1
```

Some nouns can be used as parts of regular expressions:

```
white           pattern matching one or more whitespace characters
```

```
owhite         " " optional whitespace
```

```
sol            " " the start of a line
```

```
eol            " " the end of a line
```

```
any            " " any character
```

Finally, some miscellaneous verbs

```
plain text     returns a regular expression matching the plain text
```

```
  plain 'dir j.*'
```

```
dir j\.\.*
```

```
pat1 between y      result is elements of y catenated together with pat1 between each
```

```
  ' *' between 'abc'
```

```
a *b *c
```

```
  ' *' between 'p1';'p2';'p3|p4'
```

```
p1 *p2 *(p3|p4)
```

```
comment nb pattern  add comment to pattern
```

Interpretation of a pattern always stops at the first null character (0{a.}). The nb verb makes use of this by catenating a null character and comment at the end of a pattern.

```
p=. rxcomp 'some digits' nb '[:digit:]]+'
```

```
rxinfo p
```

```
+-----+
```

```
|1|[:digit:]]+ NB. some digits|
```

```
+-----+
```

```
setchars setpat    returns list of characters matching a set pattern
```

```
  setchars '[a-d[:digit:]]'
```

```
0123456789abcd
```

Character classes

The following nouns are strings which are used **within** sets to specify a character class:

alnum, alpha, blank, cntrl, digit, graph,
lower, print, punct, space, upper, xdigit

For example,

```
alpha=. '[:alpha:]'
```

Corresponding nouns, named with a leading uppercase, are patterns specifying a **set** of the character class, for example,

```
Alpha=. '[:alpha:]' NB. (same as set alpha)
```

J patterns

The following nouns, defined in `packages\regex\regj.ijs`, are patterns which match elements of J code:

Jname	matches a J name
Jnumitem, Jnum	matches a J numeric item or array (constant)
Jchar	matches a J character string
Jconst	matches a J numeric or character constant, include a. and a:
Jgassign, Jlassign, Jassign	matches J global, local, or either assignment
Jlpar, Jrpar	match J's left and right parentheses
Jsol, Jeol	match the start or end of a J sentence (ignores leading blanks and trailing blanks/comments)

Regular Expression Demo

This program demonstrates and allows you to experiment with regular expressions. Some standard text can be searched, or you can open any text file which will be displayed. When you type in a pattern and hit the Match button, the text will be searched for that pattern. All matches will be displayed in red and underscored.

A set of canned patterns can be tried by selecting them in the Patterns menu.

Run this demo from the **Studio|Demos** menu command. This demo requires a richeditm control and is only supported in the J Win95 and NT versions.

Regular Expression Copyright

The J system and its interface to the regular expression pattern matching library is
Copyright (©) 1994-1998 Iverson Software Inc.

The regular expression library used in the interpreter is
Copyright (©) 1992, 1993, 1994, 1995 Free Software Foundation, Inc.
This library is being used according to the GNU Library Public License.

GNU Library Public License

GNU LIBRARY GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA

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

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, 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 this service 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 make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in

effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either

verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special

exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. 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 not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

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

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library 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.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), 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 distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library 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 specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "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 LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY 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 LIBRARY (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 LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS