Perl Table of Contents

1 SYNOPSIS
3 DESCRIPTION
4 AVAILABILITY
5 ENVIRONMENT
5 AUTHOR
5 FILES
5 SEE ALSO
5 DIAGNOSTICS
5 BUGS
5 NOTES
6 DESCRIPTION
6 Supported Environments
6 Core Changes
6 List assignment to %ENV works
6 Change to
6 Compilation option: Binary compatibility with 5.003
6 $PERL5OPT environment variable
6 Limitations on -M, -m, and -T options
6 More precise warnings
6 Deprecated: Inherited AUTOLOAD for non-methods
7 Previously deprecated %OVERLOAD is no longer usable
7 Subroutine arguments created only when they're modified
7 Group vector changeable with $)
7 Fixed parsing of $$<digit>, &$<digit>, etc.
7 Fixed localization of $<digit>, $&, etc.
8 No resetting of $. on implicit close
8 wantarray may return undef
8 eval EXPR determines value of EXPR in scalar context
8 Changes to tainting checks
8 New Opcode module and revised Safe module
8 Embedding improvements
9 Internal change: FileHandle class based on IO::* classes
9 Internal change: PerlIO abstraction interface
9 New and changed syntax
9 New and changed builtin constants
9 New and changed builtin variables
9 New and changed builtin functions
12 New builtin methods
13 TIEHANDLE now supported
14 Malloc enhancements
15 Miscellaneous efficiency enhancements
15 Support for More Operating Systems
15 Win32
15 Plan 9
15 QNX
15 AmigaOS
15 Pragmata
16 Modules
16 Required Updates
16 Installation directories
17 Module information summary
17 Fcntl
18 IO

Perl Table of Contents

18 Math::Complex
18 Math::Trig
18 DB_File
19 Net::Ping
19 Object-oriented overrides for builtin operators
19 Utility Changes
19 pod2html
19 xsubpp
20 C Language API Changes
20 Documentation Changes
20 New Diagnostics
25 BUGS
26 SEE ALSO
26 HISTORY
27 DESCRIPTION
27 About the new versioning system
27 Incompatible Changes
27 WARNING: This version is not binary compatible with Perl 5.004.
27 Default installation structure has changed
27 Perl Source Compatibility
27 C Source Compatibility
28 Binary Compatibility
28 Security fixes may affect compatibility
28 Relaxed new mandatory warnings introduced in 5.004
28 Licensing
28 Core Changes
28 Threads
29 Compiler
29 Regular Expressions
30 Improved malloc()
30 Quicksort is internally implemented
31 Reliable signals
31 Reliable stack pointers
31 More generous treatment of carriage returns
31 Memory leaks
31 Better support for multiple interpreters
31 Behavior of local() on array and hash elements is now well-defined
31 %! is transparently tied to the the Errno manpage module
31 Pseudo-hashes are supported
31 EXPR foreach EXPR is supported
31 Keywords can be globally overridden
31 $^E is meaningful on Win32
31 foreach (1..1000000) optimized
31 Foo:: can be used as implicitly quoted package name
31 exists $Foo::{Bar::} tests existence of a package
32 Better locale support
32 Experimental support for 64-bit platforms
32 prototype() returns useful results on builtins
32 Extended support for exception handling
32 Re-blessing in DESTROY() supported for chaining DESTROY() methods
32 All printf format conversions are handled internally
32 New INIT keyword
32 New lock keyword
32 New qr// operator
32 our is now a reserved word

Perl Table of Contents

32 Tied arrays are now fully supported
32 Tied handles support is better
32 4th argument to substr
32 Negative LENGTH argument to splice
33 Magic lvalues are now more magical
33 <> now reads in records
33 Supported Platforms
33 New Platforms
33 Changes in existing support
33 Modules and Pragmata
33 New Modules
34 Changes in existing modules
35 Utility Changes
35 Documentation Changes
35 New Diagnostics
38 Obsolete Diagnostics
38 Configuration Changes
38 BUGS
38 SEE ALSO
39 HISTORY
40 DESCRIPTION
73 AUTHORS
74 SEE ALSO
75 SYNOPSIS
76 DESCRIPTION
77 Co-existence with stdio
79 DESCRIPTION
80 DESCRIPTION
80 If we could talk to the animals...
81 Introducing the method invocation arrow
81 Invoking a barnyard
82 The extra parameter of method invocation
82 Calling a second method to simplify things
83 Inheriting the windpipes
83 A few notes about @ISA
84 Overriding the methods
85 Starting the search from a different place
85 The SUPER way of doing things
85 Where we're at so far...
86 A horse is a horse, of course of course or is it?
87 Invoking an instance method
87 Accessing the instance data
87 How to build a horse
88 Inheriting the constructor
89 Making a method work with either classes or instances
89 Adding parameters to a method
90 More interesting instances
91 A horse of a different color
92 Summary
92 SEE ALSO
92 COPYRIGHT
93 DESCRIPTION
93 OO SCALING TIPS
93 INSTANCE VARIABLES
94 SCALAR INSTANCE VARIABLES

Perl Table of Contents

94 INSTANCE VARIABLE INHERITANCE
95 OBJECT RELATIONSHIPS
95 OVERRIDING SUPERCLASS METHODS
96 USING RELATIONSHIP WITH SDBM
97 THINKING OF CODE REUSE
99 CLASS CONTEXT AND THE OBJECT
100 INHERITING A CONSTRUCTOR
101 DELEGATION
102 DESCRIPTION
102 THE CALL_ FUNCTIONS
103 FLAG VALUES
103 G_VOID
103 G_SCALAR
104 G_ARRAY
104 G_DISCARD
104 G_NOARGS
105 G_EVAL
105 G_KEEPERR
105 Determining the Context
106 KNOWN PROBLEMS
106 EXAMPLES
107 No Parameters, Nothing returned
107 Passing Parameters
109 Returning a Scalar
110 Returning a list of values
112 Returning a list in a scalar context
112 Returning Data from Perl via the parameter list
113 Using G_EVAL
115 Using G_KEEPERR
116 Using call_sv
118 Using call_argv
119 Using call_method
120 Using GIMME_V
121 Using Perl to dispose of temporaries
122 Strategies for storing Callback Context Information
128 Alternate Stack Manipulation
129 Creating and calling an anonymous subroutine in C
129 SEE ALSO
129 AUTHOR
129 DATE
130 DESCRIPTION
130 Layout
131 Using The Back Ends
131 The Cross Referencing Back End
131 The Decompiling Back End
132 The Lint Back End
132 The Simple C Back End
133 The Bytecode Back End
133 The Optimized C Back End
135 KNOWN PROBLEMS
135 AUTHOR
136 DESCRIPTION
136 Variable names
137 Context
137 Scalar values

Perl Table of Contents

139 Scalar value constructors
142 List value constructors
145 Slices
146 Typeglobs and Filehandles
147 SEE ALSO
148 SYNOPSIS
148 DESCRIPTION
148 The Filter
148 An Example the NULL termination problem.
149 Another Example Key is a C int.
150 SEE ALSO
150 AUTHOR
151 DESCRIPTION
151 Debugger Internals
152 Writing Your Own Debugger
152 Frame Listing Output Examples
156 Debugging regular expressions
156 Compile-time output
157 Types of nodes
160 Run-time output
160 Debugging Perl memory usage
161 Using $ENV{PERL_DEBUG_MSTATS}
162 Example of using -DL switch
163 -DL details
163 Limitations of -DL statistics
163 SEE ALSO
164 DESCRIPTION
164 The Perl Debugger
164 Debugger Commands
168 Configurable Options
170 Debugger input/output
172 Debugging compile-time statements
172 Debugger Customization
173 Readline Support
173 Editor Support for Debugging
173 The Perl Profiler
173 Debugging regular expressions
173 Debugging memory usage
173 SEE ALSO
174 BUGS
175 DESCRIPTION
175 Summary of changes between 5.6.0 and 5.6.1
175 Security Issues
175 Core bug fixes
178 Core features
178 Configuration issues
178 Documentation
179 Bundled modules
180 Platform-specific improvements
181 Core Enhancements
181 Interpreter cloning, threads, and concurrency
182 Lexically scoped warning categories
182 Unicode and UTF-8 support
182 Support for interpolating named characters
182

Perl Table of Contents

182 Support for strings represented as a vector of ordinals
183 Improved Perl version numbering system
183 New syntax for declaring subroutine attributes
184 File and directory handles can be autovivified
184 open() with more than two arguments
184 64-bit support
185 Large file support
185 Long doubles
185
186 Enhanced support for sort() subroutines
186 sort $coderef @foo allowed
186 File globbing implemented internally
186 Support for CHECK blocks
186 POSIX character class syntax [: :] supported
186 Better pseudo-random number generator
186 Improved qw// operator
186 Better worst-case behavior of hashes
186 pack() format 'Z' supported
186 pack() format modifier '!' supported
187 pack() and unpack() support counted strings
187 Comments in pack() templates
187 Weak references
187 Binary numbers supported
187 Lvalue subroutines
187 Some arrows may be omitted in calls through references
187 Boolean assignment operators are legal lvalues
187 exists() is supported on subroutine names
188 exists() and delete() are supported on array elements
188 Pseudo-hashes work better
188 Automatic flushing of output buffers
188 Better diagnostics on meaningless filehandle operations
188 Where possible, buffered data discarded from duped input filehandle
189 eof() has the same old magic as <>
189 binmode() can be used to set :crlf and :raw modes
189 -T filetest recognizes UTF-8 encoded files as
189 system(), backticks and pipe open now reflect exec() failure
189 Improved diagnostics
189 Diagnostics follow STDERR
189 More consistent close-on-exec behavior
190 syswrite() ease-of-use
190 Better syntax checks on parenthesized unary operators
190 Bit operators support full native integer width
190 Improved security features
190 More functional bareword prototype (*)
190 require and do may be overridden
190 $^X variables may now have names longer than one character
191 New variable $^C reflects -c switch
191 New variable $^V contains Perl version as a string
191 Optional Y2K warnings
191 Arrays now always interpolate into double-quoted strings
191 Modules and Pragmata
191 Modules
198 Pragmata
198 Utility Changes
198 dprofpp

Perl Table of Contents

198 find2perl
198 h2xs
198 perlcc
198 perldoc
198 The Perl Debugger
198 Improved Documentation
199 Performance enhancements
199 Simple sort() using { $a <=> $b } and the like are optimized
199 Optimized assignments to lexical variables
200 Faster subroutine calls
200 delete(), each(), values() and hash iteration are faster
200 Installation and Configuration Improvements
200 -Dusethreads means something different
200 New Configure flags
200 Threadedness and 64-bitness now more daring
200 Long Doubles
201 -Dusemorebits
201 -Duselargefiles
201 installusrbinperl
201 SOCKS support
201 -A flag
201 Enhanced Installation Directories
201 gcc automatically tried if 'cc' does not seem to be working
201 Platform specific changes
201 Supported platforms
202 DOS
202 OS390 (OpenEdition MVS)
202 VMS
202 Win32
203 Significant bug fixes
203 <HANDLE> on empty files
203 eval '...' improvements
203 All compilation errors are true errors
203 Implicitly closed filehandles are safer
203 Behavior of list slices is more consistent
204 ($) prototype and $foo{a}
204 goto &sub and AUTOLOAD
204 -bareword allowed under use integer
204 Failures in DESTROY()
204 Locale bugs fixed
204 Memory leaks
204 Spurious subroutine stubs after failed subroutine calls
204 Taint failures under -U
204 END blocks and the -c switch
205 Potential to leak DATA filehandles
205 New or Changed Diagnostics
212 New tests
212 Incompatible Changes
212 Perl Source Incompatibilities
214 C Source Incompatibilities
215 Compatible C Source API Changes
215 Binary Incompatibilities
215 Known Problems
216 Localizing a tied hash element may leak memory
216 Known test failures

216 EBCDIC platforms not fully supported
216 UNICOS/mk CC failures during Configure run
217 Arrow operator and arrays
217 Experimental features
217 Obsolete Diagnostics
218 Reporting Bugs
218 SEE ALSO
218 HISTORY
219 DESCRIPTION
264 DESCRIPTION
264 REFERENCES
265 COMMON MISTAKES
267 CAVEAT ON PRECEDENCE
267 WHY YOU SHOULD ALWAYS use strict
268 DEBUGGING
268 CODE EXAMPLES
268 ARRAYS OF ARRAYS
268 Declaration of a ARRAY OF ARRAYS
268 Generation of a ARRAY OF ARRAYS
269 Access and Printing of a ARRAY OF ARRAYS
269 HASHES OF ARRAYS
269 Declaration of a HASH OF ARRAYS
269 Generation of a HASH OF ARRAYS
270 Access and Printing of a HASH OF ARRAYS
271 ARRAYS OF HASHES
271 Declaration of a ARRAY OF HASHES
271 Generation of a ARRAY OF HASHES
272 Access and Printing of a ARRAY OF HASHES
272 HASHES OF HASHES
273 Declaration of a HASH OF HASHES
273 Generation of a HASH OF HASHES
274 Access and Printing of a HASH OF HASHES
275 MORE ELABORATE RECORDS
275 Declaration of MORE ELABORATE RECORDS
275 Declaration of a HASH OF COMPLEX RECORDS
276 Generation of a HASH OF COMPLEX RECORDS
277 Database Ties
278 SEE ALSO
278 AUTHOR
279 DESCRIPTION
279 PREAMBLE
279 ROADMAP
279 Compiling your C program
281 Adding a Perl interpreter to your C program
281 Calling a Perl subroutine from your C program
282 Evaluating a Perl statement from your C program
284 Performing Perl pattern matches and substitutions from your C program
287 Fiddling with the Perl stack from your C program
289 Maintaining a persistent interpreter
292 Maintaining multiple interpreter instances
294 Using Perl modules, which themselves use C libraries, from your C program
295 Embedding Perl under Win32
295 MORAL
295 AUTHOR
295 COPYRIGHT

Perl Table of Contents

296 DESCRIPTION
296 perlfaq: Structural overview of the FAQ.
296 the perlfaq1 manpage: General Questions About Perl
296 the perlfaq2 manpage: Obtaining and Learning about Perl
297 the perlfaq3 manpage: Programming Tools
298 the perlfaq4 manpage: Data Manipulation
300 the perlfaq5 manpage: Files and Formats
301 the perlfaq6 manpage: Regexps
302 the perlfaq7 manpage: General Perl Language Issues
302 the perlfaq8 manpage: System Interaction
304 the perlfaq9 manpage: Networking
305 About the perlfaq documents
305 Where to get the perlfaq
305 How to contribute to the perlfaq
305 What will happen if you mail your Perl programming problems to the authors
305 Credits
305 Author and Copyright Information
305 Bundled Distributions
305 Disclaimer
305 Changes
307 DESCRIPTION
307 What is Perl?
307 Who supports Perl? Who develops it? Why is it free?
307 Which version of Perl should I use?
307 What are perl4 and perl5?
308 What is perl6?
308 How stable is Perl?
308 Is Perl difficult to learn?
308 How does Perl compare with other languages like Java, Python, REXX, Scheme, or Tcl?
309 Can I do [task] in Perl?
309 When shouldn't I program in Perl?
309 What's the difference between
309 Is it a Perl program or a Perl script?
310 What is a JAPH?
310 Where can I get a list of Larry Wall witticisms?
310 How can I convince my sysadmin/supervisor/employees to use version 5/5.005/Perl instead of some other language?
311 AUTHOR AND COPYRIGHT
312 DESCRIPTION
312 What machines support Perl? Where do I get it?
312 How can I get a binary version of Perl?
312 I don't have a C compiler on my system. How can I compile perl?
312 I copied the Perl binary from one machine to another, but scripts don't work.
313 I grabbed the sources and tried to compile but gdbm/dynamic loading/malloc/linking/... failed. How do I make it wo
313 What modules and extensions are available for Perl? What is CPAN? What does CPAN/src/... mean?
313 Is there an ISO or ANSI certified version of Perl?
313 Where can I get information on Perl?
314 What are the Perl newsgroups on Usenet? Where do I post questions?
314 Where should I post source code?
314 Perl Books
317 Perl in Magazines
317 Perl on the Net: FTP and WWW Access
317 What mailing lists are there for Perl?
318 Archives of comp.lang.perl.misc
318 Where can I buy a commercial version of Perl?
319 Where do I send bug reports?

Perl Table of Contents

319 What is perl.com? Perl Mongers? pm.org? perl.org?
319 AUTHOR AND COPYRIGHT
320 DESCRIPTION
320 How do I do (anything)?
320 How can I use Perl interactively?
320 Is there a Perl shell?
320 How do I debug my Perl programs?
321 How do I profile my Perl programs?
321 How do I cross-reference my Perl programs?
321 Is there a pretty-printer (formatter) for Perl?
322 Is there a ctags for Perl?
322 Is there an IDE or Windows Perl Editor?
324 Where can I get Perl macros for vi?
324 Where can I get perl-mode for emacs?
324 How can I use curses with Perl?
324 How can I use X or Tk with Perl?
324 How can I generate simple menus without using CGI or Tk?
324 What is undump?
325 How can I make my Perl program run faster?
325 How can I make my Perl program take less memory?
325 Is it unsafe to return a pointer to local data?
326 How can I free an array or hash so my program shrinks?
326 How can I make my CGI script more efficient?
327 How can I hide the source for my Perl program?
327 How can I compile my Perl program into byte code or C?
327 How can I compile Perl into Java?
328 How can I get #!perl to work on [MS-DOS,NT,...]?
328 Can I write useful Perl programs on the command line?
328 Why don't Perl one-liners work on my DOS/Mac/VMS system?
329 Where can I learn about CGI or Web programming in Perl?
330 Where can I learn about object-oriented Perl programming?
330 Where can I learn about linking C with Perl? [h2xs, xsubpp]
330 I've read perlembed, perlguts, etc., but I can't embed perl in my C program; what am I doing wrong?
330 When I tried to run my script, I got this message. What does it mean?
330 What's MakeMaker?
330 AUTHOR AND COPYRIGHT
331 DESCRIPTION
331 Data: Numbers
331 Why am I getting long decimals (eg, 19.9499999999999) instead of the numbers I should be getting (eg, 19.95)?
331 Why isn't my octal data interpreted correctly?
331 Does Perl have a round() function? What about ceil() and floor()? Trig functions?
332 How do I convert bits into ints?
332 Why doesn't & work the way I want it to?
333 How do I multiply matrices?
333 How do I perform an operation on a series of integers?
333 How can I output Roman numerals?
333 Why aren't my random numbers random?
334 Data: Dates
334 How do I find the week-of-the-year/day-of-the-year?
334 How do I find the current century or millennium?
334 How can I compare two dates and find the difference?
335 How can I take a string and turn it into epoch seconds?
335 How can I find the Julian Day?
335 How do I find yesterday's date?
336 Does Perl have a Year 2000 problem? Is Perl Y2K compliant?

Perl Table of Contents

337 Data: Strings
337 How do I validate input?
337 How do I unescape a string?
337 How do I remove consecutive pairs of characters?
337 How do I expand function calls in a string?
337 How do I find matching/nesting anything?
338 How do I reverse a string?
338 How do I expand tabs in a string?
338 How do I reformat a paragraph?
338 How can I access/change the first N letters of a string?
339 How do I change the Nth occurrence of something?
339 How can I count the number of occurrences of a substring within a string?
339 How do I capitalize all the words on one line?
340 How can I split a [character] delimited string except when inside [character]? (Comma-separated files)
341 How do I strip blank space from the beginning/end of a string?
341 How do I pad a string with blanks or pad a number with zeroes?
342 How do I extract selected columns from a string?
342 How do I find the soundex value of a string?
342 How can I expand variables in text strings?
343 What's wrong with always quoting
343 Why don't my <<HERE documents work?
344 Data: Arrays
345 What is the difference between a list and an array?
345 What is the difference between $array[1] and @array[1]?
345 How can I remove duplicate elements from a list or array?
346 How can I tell whether a list or array contains a certain element?
347 How do I compute the difference of two arrays? How do I compute the intersection of two arrays?
347 How do I test whether two arrays or hashes are equal?
348 How do I find the first array element for which a condition is true?
348 How do I handle linked lists?
349 How do I handle circular lists?
349 How do I shuffle an array randomly?
350 How do I process/modify each element of an array?
350 How do I select a random element from an array?
350 How do I permute N elements of a list?
351 How do I sort an array by (anything)?
352 How do I manipulate arrays of bits?
353 Why does defined() return true on empty arrays and hashes?
353 Data: Hashes (Associative Arrays)
353 How do I process an entire hash?
354 What happens if I add or remove keys from a hash while iterating over it?
354 How do I look up a hash element by value?
354 How can I know how many entries are in a hash?
354 How do I sort a hash (optionally by value instead of key)?
355 How can I always keep my hash sorted?
355 What's the difference between
356 Why don't my tied hashes make the defined/exists distinction?
356 How do I reset an each() operation part-way through?
356 How can I get the unique keys from two hashes?
357 How can I store a multidimensional array in a DBM file?
357 How can I make my hash remember the order I put elements into it?
357 Why does passing a subroutine an undefined element in a hash create it?
357 How can I make the Perl equivalent of a C structure/ class/hash or array of hashes or arrays?
358 How can I use a reference as a hash key?
358 Data: Misc

358 How do I handle binary data correctly?
358 How do I determine whether a scalar is a number/whole/integer/float?
359 How do I keep persistent data across program calls?
359 How do I print out or copy a recursive data structure?
359 How do I define methods for every class/object?
360 How do I verify a credit card checksum?
360 How do I pack arrays of doubles or floats for XS code?
360 AUTHOR AND COPYRIGHT
361 DESCRIPTION
361 How do I flush/unbuffer an output filehandle? Why must I do this?
362 How do I change one line in a file/delete a line in a file/insert a line in the middle of a file/append to the beginning of
363 How do I count the number of lines in a file?
363 How do I make a temporary file name?
364 How can I manipulate fixed-record-length files?
365 How can I make a filehandle local to a subroutine? How do I pass filehandles between subroutines? How do I make
366 How can I use a filehandle indirectly?
367 How can I set up a footer format to be used with write()?
367 How can I write() into a string?
367 How can I output my numbers with commas added?
368 How can I translate tildes (~) in a filename?
368 How come when I open a file read-write it wipes it out?
369 Why do I sometimes get an
370 Is there a leak/bug in glob()?
370 How can I open a file with a leading
370 How can I reliably rename a file?
371 How can I lock a file?
371 Why can't I just open(FH,
371 I still don't get locking. I just want to increment the number in the file. How can I do this?
372 How do I randomly update a binary file?
372 How do I get a file's timestamp in perl?
372 How do I set a file's timestamp in perl?
373 How do I print to more than one file at once?
373 How can I read in an entire file all at once?
374 How can I read in a file by paragraphs?
374 How can I read a single character from a file? From the keyboard?
376 How can I tell whether there's a character waiting on a filehandle?
377 How do I do a tail -f in perl?
378 How do I dup() a filehandle in Perl?
378 How do I close a file descriptor by number?
378 Why can't I use
378 Why doesn't glob(
378 Why does Perl let me delete read-only files? Why does -i clobber protected files? Isn't this a bug in Perl?
379 How do I select a random line from a file?
379 Why do I get weird spaces when I print an array of lines?
379 AUTHOR AND COPYRIGHT
380 DESCRIPTION
380 How can I hope to use regular expressions without creating illegible and unmaintainable code?
381 I'm having trouble matching over more than one line. What's wrong?
381 How can I pull out lines between two patterns that are themselves on different lines?
382 I put a regular expression into $/ but it didn't work. What's wrong?
382 How do I substitute case insensitively on the LHS while preserving case on the RHS?
383 How can I make w match national character sets?
383 How can I match a locale-smart version of /[a-zA-Z]/?
384 How can I quote a variable to use in a regex?
384 What is /o really for?

Perl Table of Contents

384 How do I use a regular expression to strip C style comments from a file?
385 Can I use Perl regular expressions to match balanced text?
386 What does it mean that regexes are greedy? How can I get around it?
386 How do I process each word on each line?
386 How can I print out a word-frequency or line-frequency summary?
387 How can I do approximate matching?
387 How do I efficiently match many regular expressions at once?
387 Why don't word-boundary searches with b work for me?
388 Why does using $&, $`, or $' slow my program down?
388 What good is G in a regular expression?
389 Are Perl regexes DFAs or NFAs? Are they POSIX compliant?
389 What's wrong with using grep or map in a void context?
389 How can I match strings with multibyte characters?
390 How do I match a pattern that is supplied by the user?
390 AUTHOR AND COPYRIGHT
392 DESCRIPTION
392 Can I get a BNF/yacc/RE for the Perl language?
392 What are all these $@%&* punctuation signs, and how do I know when to use them?
392 Do I always/never have to quote my strings or use semicolons and commas?
393 How do I skip some return values?
393 How do I temporarily block warnings?
393 What's an extension?
393 Why do Perl operators have different precedence than C operators?
394 How do I declare/create a structure?
394 How do I create a module?
396 How do I create a class?
396 How can I tell if a variable is tainted?
396 What's a closure?
397 What is variable suicide and how can I prevent it?
397 How can I pass/return a {Function, FileHandle, Array, Hash, Method, Regex}?
399 How do I create a static variable?
400 What's the difference between dynamic and lexical (static) scoping? Between local() and my()?
400 How can I access a dynamic variable while a similarly named lexical is in scope?
401 What's the difference between deep and shallow binding?
401 Why doesn't
401 How do I redefine a builtin function, operator, or method?
401 What's the difference between calling a function as &foo and foo()?
402 How do I create a switch or case statement?
403 How can I catch accesses to undefined variables/functions/methods?
404 Why can't a method included in this same file be found?
404 How can I find out my current package?
404 How can I comment out a large block of perl code?
405 How do I clear a package?
405 How can I use a variable as a variable name?
406 AUTHOR AND COPYRIGHT
408 DESCRIPTION
408 How do I find out which operating system I'm running under?
408 How come exec() doesn't return?
408 How do I do fancy stuff with the keyboard/screen/mouse?
408 How do I print something out in color?
409 How do I read just one key without waiting for a return key?
410 How do I check whether input is ready on the keyboard?
410 How do I clear the screen?
411 How do I get the screen size?
411 How do I ask the user for a password?

Perl Table of Contents

411 How do I read and write the serial port?
413 How do I decode encrypted password files?
413 How do I start a process in the background?
414 How do I trap control characters/signals?
414 How do I modify the shadow password file on a Unix system?
414 How do I set the time and date?
414 How can I sleep() or alarm() for under a second?
415 How can I measure time under a second?
415 How can I do an atexit() or setjmp()/longjmp()? (Exception handling)
416 Why doesn't my sockets program work under System V (Solaris)? What does the error message
416 How can I call my system's unique C functions from Perl?
416 Where do I get the include files to do ioctl() or syscall()?
416 Why do setuid perl scripts complain about kernel problems?
416 How can I open a pipe both to and from a command?
417 Why can't I get the output of a command with system()?
417 How can I capture STDERR from an external command?
418 Why doesn't open() return an error when a pipe open fails?
418 What's wrong with using backticks in a void context?
419 How can I call backticks without shell processing?
419 Why can't my script read from STDIN after I gave it EOF (^D on Unix, ^Z on MS-DOS)?
420 How can I convert my shell script to perl?
420 Can I use perl to run a telnet or ftp session?
420 How can I write expect in Perl?
420 Is there a way to hide perl's command line from programs such as
420 I {changed directory, modified my environment} in a perl script. How come the change disappeared when I exited th
421 How do I close a process's filehandle without waiting for it to complete?
421 How do I fork a daemon process?
421 How do I find out if I'm running interactively or not?
421 How do I timeout a slow event?
422 How do I set CPU limits?
422 How do I avoid zombies on a Unix system?
422 How do I use an SQL database?
422 How do I make a system() exit on control-C?
422 How do I open a file without blocking?
422 How do I install a module from CPAN?
423 What's the difference between require and use?
423 How do I keep my own module/library directory?
423 How do I add the directory my program lives in to the module/library search path?
423 How do I add a directory to my include path at runtime?
424 What is socket.ph and where do I get it?
424 AUTHOR AND COPYRIGHT
425 DESCRIPTION
425 My CGI script runs from the command line but not the browser. (500 Server Error)
425 How can I get better error messages from a CGI program?
426 How do I remove HTML from a string?
427 How do I extract URLs?
427 How do I download a file from the user's machine? How do I open a file on another machine?
427 How do I make a pop-up menu in HTML?
427 How do I fetch an HTML file?
428 How do I automate an HTML form submission?
428 How do I decode or create those %-encodings on the web?
429 How do I redirect to another page?
429 How do I put a password on my web pages?
429 How do I edit my .htpasswd and .htgroup files with Perl?
429 How do I make sure users can't enter values into a form that cause my CGI script to do bad things?

430 How do I parse a mail header?
430 How do I decode a CGI form?
430 How do I check a valid mail address?
431 How do I decode a MIME/BASE64 string?
431 How do I return the user's mail address?
432 How do I send mail?
432 How do I use MIME to make an attachment to a mail message?
433 How do I read mail?
433 How do I find out my hostname/domainname/IP address?
434 How do I fetch a news article or the active newsgroups?
434 How do I fetch/put an FTP file?
434 How can I do RPC in Perl?
434 AUTHOR AND COPYRIGHT
435 DESCRIPTION
435 CONCEPTS
435 USING FILTERS
437 WRITING A SOURCE FILTER
437 WRITING A SOURCE FILTER IN C
437 CREATING A SOURCE FILTER AS A SEPARATE EXECUTABLE
438 WRITING A SOURCE FILTER IN PERL
439 USING CONTEXT: THE DEBUG FILTER
442 CONCLUSION
442 REQUIREMENTS
442 AUTHOR
443 Copyrights
444 SYNOPSIS
444 DESCRIPTION
444 Behavior of other Perl features in forked pseudo-processes
445 Resource limits
445 Killing the parent process
445 Lifetime of the parent process and pseudo-processes
445 CAVEATS AND LIMITATIONS
447 BUGS
448 AUTHOR
448 SEE ALSO
449 DESCRIPTION
451 Format Variables
452 NOTES
453 Footers
453 Accessing Formatting Internals
454 WARNINGS
455 DESCRIPTION
456 Perl Functions by Category
456 Functions for SCALARs or strings
456 Regular expressions and pattern matching
456 Numeric functions
456 Functions for real @ARRAYs
456 Functions for list data
456 Functions for real %HASHes
456 Input and output functions
456 Functions for fixed length data or records
456 Functions for filehandles, files, or directories
456 Keywords related to the control flow of your perl program
456 Keywords related to scoping
456 Miscellaneous functions

Perl Table of Contents

456 Functions for processes and process groups
456 Keywords related to perl modules
456 Keywords related to classes and object-orientedness
456 Low-level socket functions
457 System V interprocess communication functions
457 Fetching user and group info
457 Fetching network info
457 Time-related functions
457 Functions new in perl5
457 Functions obsoleted in perl5
457 Portability
457 Alphabetical Listing of Perl Functions
457 -X FILEHANDLE
457 -X EXPR
457 -X
459 abs VALUE
459 abs
459 accept NEWSOCKET,GENERICSOCKET
459 alarm SECONDS
459 alarm
460 atan2 Y,X
460 bind SOCKET,NAME
460 binmode FILEHANDLE, DISCIPLINE
460 binmode FILEHANDLE
461 bless REF,CLASSNAME
461 bless REF
461 caller EXPR
461 caller
462 chdir EXPR
462 chmod LIST
462 chomp VARIABLE
462 chomp LIST
462 chomp
463 chop VARIABLE
463 chop LIST
463 chop
463 chown LIST
463 chr NUMBER
463 chr
464 chroot FILENAME
464 chroot
464 close FILEHANDLE
464 close
464 closedir DIRHANDLE
464 connect SOCKET,NAME
464 continue BLOCK
465 cos EXPR
465 cos
465 crypt PLAINTEXT,SALT
466 dbmclose HASH
466 dbmopen HASH,DBNAME,MASK
466 defined EXPR
466 defined
467 delete EXPR
468 die LIST

469 do BLOCK
469 do SUBROUTINE(LIST)
469 do EXPR
470 dump LABEL
470 dump
470 each HASH
471 eof FILEHANDLE
471 eof ()
471 eof
472 eval EXPR
472 eval BLOCK
473 exec LIST
473 exec PROGRAM LIST
474 exists EXPR
475 exit EXPR
476 exp EXPR
476 exp
476 fcntl FILEHANDLE,FUNCTION,SCALAR
476 fileno FILEHANDLE
476 flock FILEHANDLE,OPERATION
477 fork
478 format
478 formline PICTURE,LIST
478 getc FILEHANDLE
478 getc
479 getlogin
479 getpeername SOCKET
479 getpgrp PID
479 getppid
479 getpriority WHICH,WHO
479 getpwnam NAME
479 getgrnam NAME
479 gethostbyname NAME
479 getnetbyname NAME
479 getprotobyname NAME
479 getpwuid UID
479 getgrgid GID
479 getservbyname NAME,PROTO
479 gethostbyaddr ADDR,ADDRTYPE
479 getnetbyaddr ADDR,ADDRTYPE
479 getprotobynumber NUMBER
479 getservbyport PORT,PROTO
479 getpwent
480 getgrent
480 gethostent
480 getnetent
480 getprotoent
480 getservent
480 setpwent
480 setgrent
480 sethostent STAYOPEN
480 setnetent STAYOPEN
480 setprotoent STAYOPEN
480 setservent STAYOPEN
480 endpwent

480 endgrent
480 endhostent
480 endnetent
480 endprotoent
480 endservent
481 getsockname SOCKET
482 getsockopt SOCKET,LEVEL,OPTNAME
482 glob EXPR
482 glob
482 gmtime EXPR
483 goto LABEL
483 goto EXPR
483 goto &NAME
483 grep BLOCK LIST
483 grep EXPR,LIST
483 hex EXPR
483 hex
484 import
484 index STR,SUBSTR,POSITION
484 index STR,SUBSTR
484 int EXPR
484 int
484 ioctl FILEHANDLE,FUNCTION,SCALAR
485 join EXPR,LIST
485 keys HASH
486 kill SIGNAL, LIST
486 last LABEL
486 last
486 lc EXPR
486 lc
486 lcfirst EXPR
486 lcfirst
486 length EXPR
486 length
487 link OLDFILE,NEWFILE
487 listen SOCKET,QUEUESIZE
487 local EXPR
487 localtime EXPR
488 lock
488 log EXPR
488 log
488 lstat FILEHANDLE
488 lstat EXPR
488 lstat
488 m//
488 map BLOCK LIST
488 map EXPR,LIST
489 mkdir FILENAME,MASK
489 mkdir FILENAME
489 msgctl ID,CMD,ARG
489 msgget KEY,FLAGS
489 msgrcv ID,VAR,SIZE,TYPE,FLAGS
489 msgsnd ID,MSG,FLAGS
490 my EXPR
490 my EXPR : ATTRIBUTES

490 next LABEL
490 next
490 no Module LIST
490 oct EXPR
490 oct
490 open FILEHANDLE,MODE,LIST
490 open FILEHANDLE,EXPR
490 open FILEHANDLE
495 opendir DIRHANDLE,EXPR
495 ord EXPR
495 ord
495 our EXPR
496 pack TEMPLATE,LIST
501 package NAMESPACE
501 package
501 pipe READHANDLE,WRITEHANDLE
501 pop ARRAY
502 pop
502 pos SCALAR
502 pos
502 print FILEHANDLE LIST
502 print LIST
502 print
502 printf FILEHANDLE FORMAT, LIST
502 printf FORMAT, LIST
502 prototype FUNCTION
503 push ARRAY,LIST
503 q/STRING/
503 qq/STRING/
503 qr/STRING/
503 qx/STRING/
503 qw/STRING/
503 quotemeta EXPR
503 quotemeta
503 rand EXPR
503 rand
503 read FILEHANDLE,SCALAR,LENGTH,OFFSET
503 read FILEHANDLE,SCALAR,LENGTH
503 readdir DIRHANDLE
504 readline EXPR
504 readlink EXPR
504 readlink
504 readpipe EXPR
504 recv SOCKET,SCALAR,LENGTH,FLAGS
504 redo LABEL
504 redo
505 ref EXPR
505 ref
506 rename OLDNAME,NEWNAME
506 require VERSION
506 require EXPR
506 require
507 reset EXPR
507 reset
507 return EXPR

507 return
507 reverse LIST
508 rewinddir DIRHANDLE
508 rindex STR,SUBSTR,POSITION
508 rindex STR,SUBSTR
508 rmdir FILENAME
508 rmdir
508 s///
508 scalar EXPR
508 seek FILEHANDLE,POSITION,WHENCE
509 seekdir DIRHANDLE,POS
509 select FILEHANDLE
509 select
509 select RBITS,WBITS,EBITS,TIMEOUT
510 semctl ID,SEMNUM,CMD,ARG
510 semget KEY,NSEMS,FLAGS
510 semop KEY,OPSTRING
511 send SOCKET,MSG,FLAGS,TO
511 send SOCKET,MSG,FLAGS
511 setpgrp PID,PGRP
511 setpriority WHICH,WHO,PRIORITY
511 setsockopt SOCKET,LEVEL,OPTNAME,OPTVAL
511 shift ARRAY
511 shift
511 shmctl ID,CMD,ARG
511 shmget KEY,SIZE,FLAGS
511 shmread ID,VAR,POS,SIZE
511 shmwrite ID,STRING,POS,SIZE
512 shutdown SOCKET,HOW
512 sin EXPR
512 sin
512 sleep EXPR
512 sleep
512 socket SOCKET,DOMAIN,TYPE,PROTOCOL
512 socketpair SOCKET1,SOCKET2,DOMAIN,TYPE,PROTOCOL
513 sort SUBNAME LIST
513 sort BLOCK LIST
513 sort LIST
515 splice ARRAY,OFFSET,LENGTH,LIST
515 splice ARRAY,OFFSET,LENGTH
515 splice ARRAY,OFFSET
515 splice ARRAY
515 split /PATTERN/,EXPR,LIMIT
515 split /PATTERN/,EXPR
515 split /PATTERN/
516 split
517 sprintf FORMAT, LIST
519 sqrt EXPR
519 sqrt
519 srand EXPR
519 srand
520 stat FILEHANDLE
520 stat EXPR
520 stat
522 study SCALAR

522 study
523 sub BLOCK
523 sub NAME
523 sub NAME BLOCK
523 substr EXPR,OFFSET,LENGTH,REPLACEMENT
523 substr EXPR,OFFSET,LENGTH
523 substr EXPR,OFFSET
523 symlink OLDFILE,NEWFILE
524 syscall LIST
524 sysopen FILEHANDLE,FILENAME,MODE
524 sysopen FILEHANDLE,FILENAME,MODE,PERMS
525 sysread FILEHANDLE,SCALAR,LENGTH,OFFSET
525 sysread FILEHANDLE,SCALAR,LENGTH
525 sysseek FILEHANDLE,POSITION,WHENCE
525 system LIST
525 system PROGRAM LIST
526 syswrite FILEHANDLE,SCALAR,LENGTH,OFFSET
526 syswrite FILEHANDLE,SCALAR,LENGTH
526 syswrite FILEHANDLE,SCALAR
526 tell FILEHANDLE
526 tell
526 telldir DIRHANDLE
526 tie VARIABLE,CLASSNAME,LIST
528 tied VARIABLE
528 time
528 times
528 tr///
528 truncate FILEHANDLE,LENGTH
529 truncate EXPR,LENGTH
529 uc EXPR
529 uc
529 ucfirst EXPR
529 ucfirst
529 umask EXPR
529 umask
529 undef EXPR
529 undef
530 unlink LIST
530 unlink
530 unpack TEMPLATE,EXPR
531 untie VARIABLE
531 unshift ARRAY,LIST
531 use Module VERSION LIST
531 use Module VERSION
531 use Module LIST
531 use Module
531 use VERSION
532 utime LIST
532 values HASH
533 vec EXPR,OFFSET,BITS
537 wait
537 waitpid PID,FLAGS
537 wantarray
538 warn LIST
538 write FILEHANDLE

538 write EXPR
538 write
539 y///
540 DESCRIPTION
540 Variables
540 Datatypes
540 What is an
540 Working with SVs
543 Offsets
543 What's Really Stored in an SV?
543 Working with AVs
544 Working with HVs
546 Hash API Extensions
546 References
547 Blessed References and Class Objects
548 Creating New Variables
548 Reference Counts and Mortality
549 Stashes and Globs
550 Double-Typed SVs
551 Magic Variables
551 Assigning Magic
552 Magic Virtual Tables
554 Finding Magic
554 Understanding the Magic of Tied Hashes and Arrays
555 Localizing changes
557 Subroutines
557 XSUBs and the Argument Stack
557 Calling Perl Routines from within C Programs
558 Memory Allocation
559 PerlIO
559 Putting a C value on Perl stack
559 Scratchpads
560 Scratchpads and recursion
560 Compiled code
560 Code tree
560 Examining the tree
562 Compile pass 1: check routines
562 Compile pass 1a: constant folding
562 Compile pass 2: context propagation
562 Compile pass 3: peephole optimization
562 Examining internal data structures with the dump functions
563 How multiple interpreters and concurrency are supported
563 Background and PERL_IMPLICIT_CONTEXT
564 So what happened to dTHR?
565 How do I use all this in extensions?
566 Should I do anything special if I call perl from multiple threads?
567 Future Plans and PERL_IMPLICIT_SYS
567 Internal Functions
568 Formatted Printing of IVs, UVs, and NVs
568 Pointer-To-Integer and Integer-To-Pointer
568 Source Documentation
569 Unicode Support
569 What is Unicode, anyway?
569 How can I recognise a UTF8 string?
569 How does UTF8 represent Unicode characters?

570 How does Perl store UTF8 strings?
571 How do I convert a string to UTF8?
571 Is there anything else I need to know?
571 AUTHORS
572 SEE ALSO
573 DESCRIPTION
575 Keeping in sync
577 Why rsync the source tree
577 Why rsync the patches
578 Submitting patches
579 Finding Your Way Around
580 Elements of the interpreter
582 Internal Variable Types
583 Op Trees
585 Stacks
587 Millions of Macros
588 Poking at Perl
588 Using a source-level debugger
589 Dumping Perl Data Structures
590 Patching
593 EXTERNAL TOOLS FOR DEBUGGING PERL
593 Rational Software's Purify
593 Purify on Unix
594 Purify on NT
595 CONCLUSION
595 AUTHOR
596 DESCRIPTION
596 INTRODUCTION
596 THE KEEPERS OF THE PUMPKIN
596 PUMPKIN?
596 THE RECORDS
602 SELECTED RELEASE SIZES
605 SELECTED PATCH SIZES
607 THE KEEPERS OF THE RECORDS
608 DESCRIPTION
609 AUTHORS
609 SEE ALSO
610 DESCRIPTION
610 Signals
612 Named Pipes
613 WARNING
613 Using open() for IPC
614 Filehandles
615 Background Processes
615 Complete Dissociation of Child from Parent
615 Safe Pipe Opens
617 Bidirectional Communication with Another Process
618 Bidirectional Communication with Yourself
619 Sockets: Client/Server Communication
620 Internet Line Terminators
620 Internet TCP Clients and Servers
623 Unix-Domain TCP Clients and Servers
625 TCP Clients with IO::Socket
626 A Simple Client
626 A Webget Client

628 Interactive Client with IO::Socket
629 TCP Servers with IO::Socket
630 UDP: Message Passing
631 SysV IPC
633 NOTES
634 BUGS
634 AUTHOR
634 SEE ALSO
635 DESCRIPTION
635 Default Warnings and Optional Warnings
635 What's wrong with -w and $^W
636 Controlling Warnings from the Command Line
637 Backward Compatibility
637 Category Hierarchy
640 Fatal Warnings
640 Reporting Warnings from a Module
642 TODO
643 SEE ALSO
643 AUTHOR
644 DESCRIPTION
644 PREPARING TO USE LOCALES
644 USING LOCALES
644 The use locale pragma
645 The setlocale function
646 Finding locales
647 LOCALE PROBLEMS
647 Temporarily fixing locale problems
648 Permanently fixing locale problems
648 Permanently fixing your system's locale configuration
648 Fixing system locale configuration
648 The localeconv function
649 LOCALE CATEGORIES
649 Category LC_COLLATE: Collation
651 Category LC_CTYPE: Character Types
651 Category LC_NUMERIC: Numeric Formatting
652 Category LC_MONETARY: Formatting of monetary amounts
652 LC_TIME
652 Other categories
652 SECURITY
654 ENVIRONMENT
655 NOTES
655 Backward compatibility
655 I18N:Collate obsolete
655 Sort speed and memory use impacts
655 write() and LC_NUMERIC
656 Freely available locale definitions
656 I18n and l10n
656 An imperfect standard
656 BUGS
656 Broken systems
656 SEE ALSO
656 HISTORY
657 DESCRIPTION
657 Declaration and Access of Arrays of Arrays
658 Growing Your Own

659 Access and Printing
660 Slices
661 SEE ALSO
661 AUTHOR
662 DESCRIPTION
662 Packages
663 Symbol Tables
665 Package Constructors and Destructors
665 Perl Classes
665 Perl Modules
668 SEE ALSO
669 DESCRIPTION
669 PREAMBLE
674 PORTABILITY
674 HEY
674 AUTHOR
675 COPYRIGHT
676 DESCRIPTION
676 THE PERL MODULE LIBRARY
676 Pragmatic Modules
677 Standard Modules
682 Extension Modules
683 CPAN
683 Africa
683 Asia
685 Central America
685 Europe
689 North America
691 Oceania
692 South America
692 Modules: Creation, Use, and Abuse
692 Guidelines for Module Creation
696 Guidelines for Converting Perl 4 Library Scripts into Modules
697 Guidelines for Reusing Application Code
697 NOTE
698 DESCRIPTION
698 Warning
698 What should I make into a module?
698 Step-by-step: Preparing the ground
699 Step-by-step: Making the module
700 Step-by-step: Distributing your module
701 AUTHOR
701 SEE ALSO
702 SYNOPSIS
702 DESCRIPTION
702 Storing numbers
703 Numeric operators and numeric conversions
703 Flavors of Perl numeric operations
704 AUTHOR
704 SEE ALSO
705 DESCRIPTION
705 An Object is Simply a Reference
706 A Class is Simply a Package
707 A Method is Simply a Subroutine
708 Method Invocation

709 WARNING
710 Default UNIVERSAL methods
711 Destructors
711 Summary
711 Two-Phased Garbage Collection
713 SEE ALSO
714 SYNOPSIS
714 DESCRIPTION
714 Terms and List Operators (Leftward)
715 The Arrow Operator
715 Auto-increment and Auto-decrement
715 Exponentiation
716 Symbolic Unary Operators
716 Binding Operators
716 Multiplicative Operators
717 Additive Operators
717 Shift Operators
717 Named Unary Operators
717 Relational Operators
718 Equality Operators
718 Bitwise And
718 Bitwise Or and Exclusive Or
718 C-style Logical And
718 C-style Logical Or
719 Range Operators
720 Conditional Operator
721 Assignment Operators
721 Comma Operator
722 List Operators (Rightward)
722 Logical Not
722 Logical And
722 Logical or and Exclusive Or
722 C Operators Missing From Perl
723 Quote and Quote-like Operators
724 Regexp Quote-Like Operators
733 Gory details of parsing quoted constructs
736 I/O Operators
739 Constant Folding
739 Bitwise String Operators
740 Integer Arithmetic
740 Floating-point Arithmetic
740 Bigger Numbers
742 DESCRIPTION
742 Open A‘ la shell
742 Simple Opens
743 Pipe Opens
743 The Minus File
743 Mixing Reads and Writes
744 Filters
745 Open A‘ la C
747 Permissions A‘ la mode
747 Obscure Open Tricks
747 Re-Opening Files (dups)
748 Dispelling the Dweomer
749 Paths as Opens

749 Single Argument Open
749 Playing with STDIN and STDOUT
750 Other I/O Issues
750 Opening Non-File Files
752 Binary Files
753 File Locking
754 SEE ALSO
754 AUTHOR and COPYRIGHT
754 HISTORY
755 DESCRIPTION
755 Verbatim Paragraph
755 Command Paragraph
757 Ordinary Block of Text
758 The Intent
758 Embedding Pods in Perl Modules
759 Common Pod Pitfalls
759 SEE ALSO
759 AUTHOR
760 DESCRIPTION
760 ISSUES
761 Newlines
762 Numbers endianness and Width
763 Files and Filesystems
764 System Interaction
765 Interprocess Communication (IPC)
765 External Subroutines (XS)
765 Standard Modules
766 Time and Date
766 Character sets and character encoding
766 Internationalisation
766 System Resources
767 Security
767 Style
767 CPAN Testers
767 PLATFORMS
767 Unix
768 DOS and Derivatives
769 MacOS
771 VMS
772 VOS
773 EBCDIC Platforms
774 Acorn RISC OS
776 Other perls
776 FUNCTION IMPLEMENTATIONS
776 Alphabetical Listing of Perl Functions
782 CHANGES
783 Supported Platforms
787 SEE ALSO
787 AUTHORS / CONTRIBUTORS
787 VERSION
788 DESCRIPTION
788 Regular Expressions
793 Extended Patterns
797 Backtracking
801 Version 8 Regular Expressions

802 Warning on 1 vs $1
802 Repeated patterns matching zero-length substring
803 Combining pieces together
804 Creating custom RE engines
804 BUGS
805 SEE ALSO
806 NOTE
806 DESCRIPTION
806 Making References
809 Using References
812 Symbolic references
812 Not-so-symbolic references
813 Pseudo-hashes: Using an array as a hash
814 Function Templates
815 WARNING
815 SEE ALSO
816 DESCRIPTION
816 Who Needs Complicated Data Structures?
816 The Solution
817 Syntax
817 Making References
817 Using References
818 An Example
819 Arrow Rule
819 Solution
820 The Rest
820 Summary
820 Credits
820 Distribution Conditions
822 DESCRIPTION
822 The Guide
822 Simple word matching
823 Using character classes
825 Matching this or that
825 Grouping things and hierarchical matching
825 Extracting matches
826 Matching repetitions
826 More matching
827 Search and replace
828 The split operator
828 BUGS
828 SEE ALSO
828 AUTHOR AND COPYRIGHT
828 Acknowledgments
829 DESCRIPTION
829 Part 1: The basics
829 Simple word matching
833 Using character classes
836 Matching this or that
836 Grouping things and hierarchical matching
837 Extracting matches
839 Matching repetitions
843 Building a regexp
845 Using regular expressions in Perl
849 Part 2: Power tools

849 More on characters, strings, and character classes
852 Compiling and saving regular expressions
853 Embedding comments and modifiers in a regular expression
854 Non-capturing groupings
854 Looking ahead and looking behind
855 Using independent subexpressions to prevent backtracking
856 Conditional expressions
857 A bit of magic: executing Perl code in a regular expression
860 Pragmas and debugging
862 BUGS
863 SEE ALSO
863 AUTHOR AND COPYRIGHT
863 Acknowledgments
864 SYNOPSIS
864 DESCRIPTION
865 #! and quoting on non-Unix systems
866 Location of Perl
866 Command Switches
873 ENVIRONMENT
876 DESCRIPTION
877 Laundering and Detecting Tainted Data
878 Switches On the
878 Cleaning Up Your Path
879 Security Bugs
880 Protecting Your Programs
880 SEE ALSO
881 DESCRIPTION
884 SYNOPSIS
884 DESCRIPTION
887 Private Variables via my()
890 Persistent Private Variables
890 Temporary Values via local()
893 Lvalue subroutines
893 Passing Symbol Table Entries (typeglobs)
894 When to Still Use local()
895 Pass by Reference
897 Prototypes
899 Constant Functions
900 Overriding Built-in Functions
901 Autoloading
902 Subroutine Attributes
903 SEE ALSO
904 DESCRIPTION
904 Declarations
904 Simple statements
905 Compound statements
906 Loop Control
908 For Loops
908 Foreach Loops
909 Basic BLOCKs and Switch Statements
912 Goto
912 PODs: Embedded Documentation
913 Plain Old Comments (Not!)
914 DESCRIPTION
914 What Is A Thread Anyway?

Perl Table of Contents

914 Threaded Program Models
914 Boss/Worker
914 Work Crew
914 Pipeline
915 Native threads
916 What kind of threads are perl threads?
916 Threadsafe Modules
916 Thread Basics
916 Basic Thread Support
917 Creating Threads
918 Giving up control
918 Waiting For A Thread To Exit
919 Errors In Threads
919 Ignoring A Thread
920 Threads And Data
920 Shared And Unshared Data
920 Thread Pitfall: Races
920 Controlling access: lock()
921 Thread Pitfall: Deadlocks
922 Queues: Passing Data Around
923 Threads And Code
923 Semaphores: Synchronizing Data Access
924 Attributes: Restricting Access To Subroutines
924 Subroutine Locks
925 Methods
926 Locking A Subroutine
926 General Thread Utility Routines
926 What Thread Am I In?
926 Thread IDs
927 Are These Threads The Same?
927 What Threads Are Running?
927 A Complete Example
929 Conclusion
929 Bibliography
929 Introductory Texts
929 OS-Related References
929 Other References
930 Acknowledgements
930 AUTHOR
930 Copyrights
931 SYNOPSIS
931 DESCRIPTION
931 Tying Scalars
933 Tying Arrays
938 Tying Hashes
943 Tying FileHandles
945 UNTIE this
945 The untie Gotcha
947 SEE ALSO
947 BUGS
947 AUTHOR
948 DESCRIPTION
948 BASIC DOCUMENTATION
948 perl - Practical Extraction and Report Language
948 perlfaq - frequently asked questions about Perl ($Date: 1999/05/23 20:38:02 $)

Perl Table of Contents

948 perlbook - Perl book information
948 perlsyn - Perl syntax
949 perldata - Perl data types
949 perlop - Perl operators and precedence
950 perlsub - Perl subroutines
950 perlfunc - Perl builtin functions
952 perlreftut - Mark's very short tutorial about references
952 perldsc - Perl Data Structures Cookbook
953 perlrequick - Perl regular expressions quick start
953 perlpod - plain old documentation
953 perlstyle - Perl style guide
953 perltrap - Perl traps for the unwary
954 perlrun - how to execute the Perl interpreter
955 perldiag - various Perl diagnostics
955 perllexwarn - Perl Lexical Warnings
955 perldebtut - Perl debugging tutorial
955 perldebug - Perl debugging
956 perlvar - Perl predefined variables
957 perllol - Manipulating Arrays of Arrays in Perl
957 perlopentut - tutorial on opening things in Perl
957 perlretut - Perl regular expressions tutorial
958 perlre - Perl regular expressions
958 perlref - Perl references and nested data structures
958 perlform - Perl formats
959 perlboot - Beginner's Object-Oriented Tutorial
959 perltoot - Tom's object-oriented tutorial for perl
960 perltootc - Tom's OO Tutorial for Class Data in Perl
960 perlobj - Perl objects
961 perlbot - Bag'o Object Tricks (the BOT)
961 perltie - how to hide an object class in a simple variable
961 perlipc - Perl interprocess communication (signals, fifos, pipes, safe subprocesses, sockets, and semaphores)
962 perlfork - Perl's fork() emulation (EXPERIMENTAL, subject to change)
962 perlnumber - semantics of numbers and numeric operations in Perl
963 perlthrtut - tutorial on threads in Perl
963 perlport - Writing portable Perl
965 perllocale - Perl locale handling (internationalization and localization)
965 perlunicode - Unicode support in Perl (EXPERIMENTAL, subject to change)
966 perlebcdic - Considerations for running Perl on EBCDIC platforms
967 perlsec - Perl security
967 perlmod - Perl modules (packages and symbol tables)
967 perlmodlib - constructing new Perl modules and finding existing ones
968 perlmodinstall - Installing CPAN Modules
968 perlnewmod - preparing a new module for distribution
968 perlfaq1 - General Questions About Perl ($Revision: 1.23 $, $Date: 1999/05/23 16:08:30 $)
969 perlfaq2 - Obtaining and Learning about Perl ($Revision: 1.32 $, $Date: 1999/10/14 18:46:09 $)
969 perlfaq3 - Programming Tools ($Revision: 1.38 $, $Date: 1999/05/23 16:08:30 $)
970 perlfaq4 - Data Manipulation ($Revision: 1.49 $, $Date: 1999/05/23 20:37:49 $)
972 perlfaq5 - Files and Formats ($Revision: 1.38 $, $Date: 1999/05/23 16:08:30 $)
973 perlfaq6 - Regexes ($Revision: 1.27 $, $Date: 1999/05/23 16:08:30 $)
973 perlfaq7 - Perl Language Issues ($Revision: 1.28 $, $Date: 1999/05/23 20:36:18 $)
974 perlfaq8 - System Interaction ($Revision: 1.39 $, $Date: 1999/05/23 18:37:57 $)
975 perlfaq9 - Networking ($Revision: 1.26 $, $Date: 1999/05/23 16:08:30 $)
975 perlcompile - Introduction to the Perl Compiler-Translator
976 perlembed - how to embed perl in your C program
976 perldebguts - Guts of Perl debugging

Perl Table of Contents

977 perlxstut, perlXStut - Tutorial for writing XSUBs
977 perlxs - XS language reference manual
978 perlclib - Internal replacements for standard C library functions
979 perlguts - Introduction to the Perl API
980 perlcall - Perl calling conventions from C
981 perlutil - utilities packaged with the Perl distribution
981 perlfilter - Source Filters
981 perldbmfilter - Perl DBM Filters
982 perlapi - autogenerated documentation for the perl public API
982 perlintern - autogenerated documentation of purely internal Perl functions
983 perlapio - perl's IO abstraction interface.
983 perltodo - Perl TO-DO List
985 perlhack - How to hack at the Perl internals
986 perlhist - the Perl history records
986 perldelta - what's new for perl v5.6
990 perl5005delta, perldelta - what's new for perl5.005
992 perl5004delta, perldelta - what's new for perl5.004
994 perlaix, README.aix - Perl version 5 on IBM Unix (AIX) systems
994 perlamiga - Perl under Amiga OS
994 perlbs2000, README.BS2000 - building and installing Perl for BS2000.
995 perlcygwin, README.cygwin - Perl for Cygwin
996 perldos - Perl under DOS, W31, W95.
996 perlepoc, README.epoc - Perl for EPOC
996 perlhpux, README.hpux - Perl version 5 on Hewlett-Packard Unix (HP-UX) systems
997 perlmachten, README.machten - Perl version 5 on Power MachTen systems
997 perlmacos, README.macos - Perl under Mac OS (Classic)
997 perlmpeix, README.mpeix - Perl/iX for HP e3000 MPE
997 SYNOPSIS
997 perlos2 - Perl under OS/2, DOS, Win0.3*, Win0.95 and WinNT.
999 perlos390, README.os390 - building and installing Perl for OS/390.
999 perlsolaris, README.solaris - Perl version 5 on Solaris systems
1000 perlvmesa, README.vmesa - building and installing Perl for VM/ESA.
1000 perlvms - VMS-specific documentation for Perl
1001 perlvos, README.vos - Perl for Stratus VOS
1001 perlwin32 - Perl under Win32
1001 PRAGMA DOCUMENTATION
1001 attrs - set/get attributes of a subroutine (deprecated)
1002 re - Perl pragma to alter regular expression behaviour
1002 attributes - get/set subroutine or variable attributes
1002 attrs - set/get attributes of a subroutine (deprecated)
1002 autouse - postpone load of modules until a function is used
1002 base - Establish IS-A relationship with base class at compile time
1002 blib - Use MakeMaker's uninstalled version of a package
1002 bytes - Perl pragma to force byte semantics rather than character semantics
1003 charnames - define character names for N{named} string literal escape.
1003 constant - Perl pragma to declare constants
1003 diagnostics - Perl compiler pragma to force verbose warning diagnostics
1003 fields - compile-time class fields
1003 filetest - Perl pragma to control the filetest permission operators
1003 integer - Perl pragma to use integer arithmetic instead of floating point
1003 less - perl pragma to request less of something from the compiler
1003 lib - manipulate @INC at compile time
1004 locale - Perl pragma to use and avoid POSIX locales for built-in operations
1004 open - perl pragma to set default disciplines for input and output
1004 ops - Perl pragma to restrict unsafe operations when compiling

Perl Table of Contents

1004 overload - Package for overloading perl operations
1005 re - Perl pragma to alter regular expression behaviour
1005 sigtrap - Perl pragma to enable simple signal handling
1005 strict - Perl pragma to restrict unsafe constructs
1005 subs - Perl pragma to predeclare sub names
1005 utf8 - Perl pragma to enable/disable UTF-8 in source code
1006 vars - Perl pragma to predeclare global variable names (obsolete)
1006 warnings - Perl pragma to control optional warnings
1006 warnings::register - warnings import function
1006 MODULE DOCUMENTATION
1006 AnyDBM_File - provide framework for multiple DBMs
1006 AutoLoader - load subroutines only on demand
1006 AutoSplit - split a package for autoloading
1006 B - The Perl Compiler
1008 B::Asmdata - Autogenerated data about Perl ops, used to generate bytecode
1008 B::Assembler - Assemble Perl bytecode
1008 B::Bblock - Walk basic blocks
1008 B::Bytecode - Perl compiler's bytecode backend
1008 B::C - Perl compiler's C backend
1009 B::CC - Perl compiler's optimized C translation backend
1009 B::Concise - Walk Perl syntax tree, printing concise info about ops
1009 B::Debug - Walk Perl syntax tree, printing debug info about ops
1009 B::Deparse - Perl compiler backend to produce perl code
1010 B::Disassembler - Disassemble Perl bytecode
1010 B::Lint - Perl lint
1010 B::O, O - Generic interface to Perl Compiler backends
1010 B::Showlex - Show lexical variables used in functions or files
1010 B::Stackobj - Helper module for CC backend
1010 B::Stash - show what stashes are loaded
1010 B::Terse - Walk Perl syntax tree, printing terse info about ops
1010 B::Xref - Generates cross reference reports for Perl programs
1011 Bblock, B::Bblock - Walk basic blocks
1011 Benchmark - benchmark running times of Perl code
1011 ByteLoader - load byte compiled perl code
1011 Bytecode, B::Bytecode - Perl compiler's bytecode backend
1011 CGI - Simple Common Gateway Interface Class
1014 CGI::Apache - Backward compatibility module for CGI.pm
1014 CGI::Carp, CGI::Carp - CGI routines for writing to the HTTPD (or other) error log
1014 CGI::Cookie - Interface to Netscape Cookies
1014 CGI::Fast - CGI Interface for Fast CGI
1015 CGI::Pretty - module to produce nicely formatted HTML code
1015 CGI::Push - Simple Interface to Server Push
1015 CGI::Switch - Backward compatibility module for defunct CGI::Switch
1015 CGI::Util - Internal utilities used by CGI module
1015 CPAN - query, download and build perl modules from CPAN sites
1016 CPAN::FirstTime - Utility for CPAN::Config file Initialization
1016 CPANox, CPAN::Nox - Wrapper around CPAN.pm without using any XS module
1016 Carp, carp - warn of errors (from perspective of caller)
1017 Carp::Heavy - Carp guts
1017 Class::Struct - declare struct-like datatypes as Perl classes
1017 Config - access Perl configuration information
1020 Cwd - get pathname of current working directory
1020 DB - programmatic interface to the Perl debugging API (draft, subject to change)
1020 DB_File - Perl5 access to Berkeley DB version 1.x
1021 Data::Dumper - stringified perl data structures, suitable for both printing and eval

Perl Table of Contents

1022 Devel::DProf - a Perl code profiler
1022 Devel::Peek - A data debugging tool for the XS programmer
1022 Devel::SelfStubber - generate stubs for a SelfLoading module
1023 DirHandle - supply object methods for directory handles
1023 Dumpvalue - provides screen dump of Perl data.
1023 DynaLoader - Dynamically load C libraries into Perl code
1023 DynaLoader::XSLoader, XSLoader - Dynamically load C libraries into Perl code
1023 English - use nice English (or awk) names for ugly punctuation variables
1023 Env - perl module that imports environment variables as scalars or arrays
1023 Errno - System errno constants
1024 Exporter - Implements default import method for modules
1024 Exporter::Heavy - Exporter guts
1024 ExtUtils::Command - utilities to replace common UNIX commands in Makefiles etc.
1024 ExtUtils::Embed - Utilities for embedding Perl in C/ applications
1025 ExtUtils::Install - install files from here to there
1025 ExtUtils::Installed - Inventory management of installed modules
1025 ExtUtils::Liblist - determine libraries to use and how to use them
1025 ExtUtils::MM_Cygwin - methods to override UN∗X behaviour in ExtUtils::MakeMaker
1025 ExtUtils::MM_OS2 - methods to override UN∗X behaviour in ExtUtils::MakeMaker
1025 ExtUtils::MM_Unix - methods used by ExtUtils::MakeMaker
1028 ExtUtils::MM_VMS - methods to override UN∗X behaviour in ExtUtils::MakeMaker
1029 ExtUtils::MM_Win32 - methods to override UN∗X behaviour in ExtUtils::MakeMaker
1030 ExtUtils::MakeMaker - create an extension Makefile
1031 ExtUtils::Manifest - utilities to write and check a MANIFEST file
1031 ExtUtils::Miniperl, writemain - write the C code for perlmain.c
1031 ExtUtils::Mkbootstrap - make a bootstrap file for use by DynaLoader
1031 ExtUtils::Mksymlists - write linker options files for dynamic extension
1031 ExtUtils::Packlist - manage .packlist files
1031 ExtUtils::testlib - add blib/∗ directories to @INC
1031 Fatal - replace functions with equivalents which succeed or die
1032 Fcntl - load the C Fcntl.h defines
1032 File::Basename, fileparse - split a pathname into pieces
1032 File::CheckTree, validate - run many filetest checks on a tree
1032 File::Compare - Compare files or filehandles
1032 File::Copy - Copy files or filehandles
1032 File::DosGlob - DOS like globbing and then some
1032 File::Find, find - traverse a file tree
1033 File::Glob - Perl extension for BSD glob routine
1033 File::Path - create or remove directory trees
1033 File::Spec - portably perform operations on file names
1033 File::Spec::Epoc - methods for Epoc file specs
1033 File::Spec::Functions - portably perform operations on file names
1033 File::Spec::Mac - File::Spec for MacOS
1034 File::Spec::OS2 - methods for OS/2 file specs
1034 File::Spec::Unix - methods used by File::Spec
1035 File::Spec::VMS - methods for VMS file specs
1035 File::Spec::Win32 - methods for Win32 file specs
1036 File::Temp - return name and handle of a temporary file safely
1036 File::stat - by-name interface to Perl's built-in stat() functions
1036 FileCache - keep more files open than the system permits
1036 FileHandle - supply object methods for filehandles
1037 FindBin - Locate directory of original perl script
1037 GDBM_File - Perl5 access to the gdbm library.
1037 Getopt::Long - Extended processing of command line options
1038 Getopt::Std, getopt - Process single-character switches with switch clustering

Perl Table of Contents

1038 I18N::Collate - compare 8-bit scalar data according to the current locale
1038 IO - load various IO modules
1038 IO::Dir - supply object methods for directory handles
1038 IO::File - supply object methods for filehandles
1038 IO::Handle - supply object methods for I/O handles
1039 IO::Pipe - supply object methods for pipes
1039 IO::Poll - Object interface to system poll call
1039 IO::Seekable - supply seek based methods for I/O objects
1039 IO::Select - OO interface to the select system call
1040 IO::Socket - Object interface to socket communications
1040 IO::Socket::INET - Object interface for AF_INET domain sockets
1040 IO::Socket::UNIX - Object interface for AF_UNIX domain sockets
1040 IO::lib::IO::Dir, IO::Dir - supply object methods for directory handles
1040 IO::lib::IO::File, IO::File - supply object methods for filehandles
1041 IO::lib::IO::Handle, IO::Handle - supply object methods for I/O handles
1041 IO::lib::IO::Pipe, IO::Pipe - supply object methods for pipes
1041 IO::lib::IO::Poll, IO::Poll - Object interface to system poll call
1041 IO::lib::IO::Seekable, IO::Seekable - supply seek based methods for I/O objects
1042 IO::lib::IO::Select, IO::Select - OO interface to the select system call
1042 IO::lib::IO::Socket, IO::Socket - Object interface to socket communications
1042 IO::lib::IO::Socket::INET, IO::Socket::INET - Object interface for AF_INET domain sockets
1042 IO::lib::IO::Socket::UNIX, IO::Socket::UNIX - Object interface for AF_UNIX domain sockets
1043 IPC::Msg - SysV Msg IPC object class
1043 IPC::Open2, open2 - open a process for both reading and writing
1043 IPC::Open3, open3 - open a process for reading, writing, and error handling
1043 IPC::Semaphore - SysV Semaphore IPC object class
1043 IPC::SysV - SysV IPC constants
1043 IPC::SysV::Msg, IPC::Msg - SysV Msg IPC object class
1044 IPC::SysV::Semaphore, IPC::Semaphore - SysV Semaphore IPC object class
1044 Math::BigFloat - Arbitrary length float math package
1044 Math::BigInt - Arbitrary size integer math package
1044 Math::Complex - complex numbers and associated mathematical functions
1044 Math::Trig - trigonometric functions
1045 NDBM_File - Tied access to ndbm files
1045 Net::Ping - check a remote host for reachability
1045 Net::hostent - by-name interface to Perl's built-in gethost*() functions
1045 Net::netent - by-name interface to Perl's built-in getnet*() functions
1045 Net::protoent - by-name interface to Perl's built-in getproto*() functions
1046 Net::servent - by-name interface to Perl's built-in getserv*() functions
1046 O - Generic interface to Perl Compiler backends
1046 ODBM_File - Tied access to odbm files
1046 Opcode - Disable named opcodes when compiling perl code
1046 Opcode::Safe, Safe - Compile and execute code in restricted compartments
1047 Opcode::ops, ops - Perl pragma to restrict unsafe operations when compiling
1047 POSIX - Perl interface to IEEE Std 1003.1
1048 Pod::Checker, podchecker() - check pod documents for syntax errors
1049 Pod::Find - find POD documents in directory trees
1049 Pod::Html - module to convert pod files to HTML
1049 Pod::InputObjects - objects representing POD input paragraphs, commands, etc.
1050 Pod::LaTeX - Convert Pod data to formatted Latex
1051 Pod::Man - Convert POD data to formatted *roff input
1052 Pod::ParseUtils - helpers for POD parsing and conversion
1053 Pod::Parser - base class for creating POD filters and translators
1054 Pod::Plainer - Perl extension for converting Pod to old style Pod.
1054 Pod::Select, podselect() - extract selected sections of POD from input

Perl Table of Contents

1054 Pod::Text - Convert POD data to formatted ASCII text
1055 Pod::Text::Color - Convert POD data to formatted color ASCII text
1055 Pod::Text::Overstrike - Convert POD data to formatted overstrike text
1055 Pod::Text::Termcap, Pod::Text::Color - Convert POD data to ASCII text with format escapes
1055 Pod::Usage, pod2usage() - print a usage message from embedded pod documentation
1055 SDBM_File - Tied access to sdbm files
1055 Safe - Compile and execute code in restricted compartments
1056 Search::Dict, look - search for key in dictionary file
1056 SelectSaver - save and restore selected file handle
1056 SelfLoader - load functions only on demand
1056 Shell - run shell commands transparently within perl
1056 Socket, sockaddr_in, sockaddr_un, inet_aton, inet_ntoa - load the C socket.h defines and structure manipulators
1056 Symbol - manipulate Perl symbols and their names
1057 Sys::Hostname - Try every conceivable way to get hostname
1057 Syslog, Sys::Syslog, openlog, closelog, setlogmask, syslog - Perl interface to the UNIX syslog calls
1057 Syslog::Syslog, Sys::Syslog, openlog, closelog, setlogmask, syslog - Perl interface to the UNIX syslog calls
1057 Term::ANSIColor - Color screen output using ANSI escape sequences
1057 Term::Cap - Perl termcap interface
1057 Term::Complete - Perl word completion module
1057 Term::ReadLine - Perl interface to various readline packages. If no real package is found, substitutes stubs instead of
1058 Test - provides a simple framework for writing test scripts
1058 Test::Harness - run perl standard test scripts with statistics
1058 Text::Abbrev, abbrev - create an abbreviation table from a list
1058 Text::ParseWords - parse text into an array of tokens or array of arrays
1058 Text::Soundex - Implementation of the Soundex Algorithm as Described by Knuth
1059 Text::Tabs expand and unexpand tabs per the unix expand and unexpand
1059 Text::Wrap - line wrapping to form simple paragraphs
1059 Thread - manipulate threads in Perl (EXPERIMENTAL, subject to change)
1059 Thread::Queue - thread-safe queues
1059 Thread::Semaphore - thread-safe semaphores
1059 Thread::Signal - Start a thread which runs signal handlers reliably
1060 Thread::Specific - thread-specific keys
1060 Tie::Array - base class for tied arrays
1060 Tie::Handle, Tie::StdHandle - base class definitions for tied handles
1060 Tie::Hash, Tie::StdHash - base class definitions for tied hashes
1060 Tie::RefHash - use references as hash keys
1060 Tie::Scalar, Tie::StdScalar - base class definitions for tied scalars
1060 Tie::SubstrHash - Fixed-table-size, fixed-key-length hashing
1061 Time::Local - efficiently compute time from local and GMT time
1061 Time::gmtime - by-name interface to Perl's built-in gmtime() function
1061 Time::localtime - by-name interface to Perl's built-in localtime() function
1061 Time::tm - internal object used by Time::gmtime and Time::localtime
1061 UNIVERSAL - base class for ALL classes (blessed references)
1061 User::grent - by-name interface to Perl's built-in getgr*() functions
1061 User::pwent - by-name interface to Perl's built-in getpw*() functions
1062 Win32 - Interfaces to some Win32 API Functions
1062 XSLoader - Dynamically load C libraries into Perl code
1062 AUXILIARY DOCUMENTATION
1062 AUTHOR
1063 DESCRIPTION
1063 Infrastructure
1063 Mailing list archives
1063 Bug tracking system
1063 Regression Tests
1064 Configure

1064 Install HTML
1064 Perl Language
1064 64-bit Perl
1064 Prototypes
1064 Perl Internals
1064 magic_setisa
1064 Garbage Collection
1065 Reliable signals
1065 Interpolated regex performance bugs
1065 Memory leaks from failed eval/regcomp
1065 Make XS easier to use
1065 Make embedded Perl easier to use
1065 Namespace cleanup
1065 MULTIPLICITY
1065 MacPerl
1065 Documentation
1066 A clear division into tutorial and reference
1066 Remove the artificial distinction between operators and functions
1066 More tutorials
1066 Include a search tool
1066 Include a locate tool
1067 Separate function manpages by default
1067 Users can't find the manpages
1067 Install ALL Documentation
1067 Outstanding issues to be documented
1068 Adapt www.linuxhq.com for Perl
1068 Replace man with a perl program
1068 Unicode tutorial
1068 Modules
1068 Update the POSIX extension to conform with the POSIX 1003.1 Edition 2
1068 Module versions
1068 New modules
1068 Profiler
1068 Tie Modules
1068 Procedural options
1068 RPC
1069 y2k localtime/gmtime
1069 Export File::Find variables
1069 Ioctl
1069 Debugger attach/detach
1069 Regular Expression debugger
1069 Alternative RE Syntax
1069 Bundled modules
1069 Expect
1069 GUI::Native
1070 Update semibroken auxiliary tools; h2ph, a2p, etc.
1070 pod2html
1070 Podchecker
1070 Tom's Wishes
1070 Webperl
1070 Mobile agents
1070 POSIX on non-POSIX
1070 Portable installations
1070 Win32 Stuff
1070 Rename new headers to be consistent with the rest

1070 Sort out the spawnvp() mess
1070 Work out DLL versioning
1070 Style-check
1070 Would be nice to have
1071 Possible pragmas
1071 'less'
1071 Optimizations
1071 constant function cache
1071 foreach(reverse...)
1071 Cache eval tree
1071 rcatmaybe
1071 Shrink opcode tables
1071 Cache hash value
1071 Optimize away @_ where possible
1071 Optimize sort by { $a <=> $b }
1071 Rewrite regexp parser for better integrated optimization
1071 Vague possibilities
1071 To Do Or Not To Do
1071 Making my() work on
1071
1072
1072
1072 Threading
1072 Modules
1072 Testing
1072 $AUTOLOAD
1072 exit/die
1072 External threads
1072 Thread::Pool
1072 thread-safety
1072 Per-thread GVs
1072 Compiler
1072 Optimization
1073 Byteperl
1073 Precompiled modules
1073 Executables
1073 Typed lexicals
1073 Win32
1073 END blocks
1073 _AUTOLOAD
1073 comppadlist
1073 Cached compilation
1073 Recently Finished Tasks
1073 Figure a way out of $^(capital letter)
1073 Filenames
1073 Foreign lines
1073 Namespace cleanup
1074 ISA.pm
1074 gettimeofday
1074 autocroak?
1075 DESCRIPTION
1075 Creating a Class
1076 Object Representation
1076 Class Interface
1077 Constructors and Instance Methods

1078 Planning for the Future: Better Constructors
1078 Destructors
1079 Other Object Methods
1080 Class Data
1081 Accessing Class Data
1082 Debugging Methods
1083 Class Destructors
1083 Documenting the Interface
1084 Aggregation
1087 Inheritance
1088 Overridden Methods
1091 Multiple Inheritance
1092 UNIVERSAL: The Root of All Objects
1092 Alternate Object Representations
1093 Arrays as Objects
1094 Closures as Objects
1095 AUTOLOAD: Proxy Methods
1095 Autoloaded Data Methods
1097 Inherited Autoloaded Data Methods
1097 Metaclassical Tools
1097 Class::Struct
1100 Data Members as Variables
1101 NOTES
1101 Object Terminology
1102 SEE ALSO
1102 AUTHOR AND COPYRIGHT
1102 COPYRIGHT
1102 Acknowledgments
1103 DESCRIPTION
1103 Class Data in a Can
1103 Class Data as Package Variables
1104 Putting All Your Eggs in One Basket
1105 Inheritance Concerns
1106 The Eponymous Meta-Object
1108 Indirect References to Class Data
1110 Monadic Classes
1113 Translucent Attributes
1116 Class Data as Lexical Variables
1116 Privacy and Responsibility
1117 File-Scoped Lexicals
1119 More Inheritance Concerns
1120 Locking the Door and Throwing Away the Key
1121 Translucency Revisited
1123 NOTES
1123 SEE ALSO
1123 AUTHOR AND COPYRIGHT
1124 ACKNOWLEDGEMENTS
1124 HISTORY
1125 DESCRIPTION
1125 Awk Traps
1126 C Traps
1126 Sed Traps
1127 Shell Traps
1127 Perl Traps
1128 Perl4 to Perl5 Traps

1128 Discontinuance, Deprecation, and BugFix traps
1132 Parsing Traps
1132 Numerical Traps
1134 General data type traps
1136 Context Traps - scalar, list contexts
1137 Precedence Traps
1139 General Regular Expression Traps using s///, etc.
1141 Subroutine, Signal, Sorting Traps
1142 OS Traps
1143 Interpolation Traps
1145 DBM Traps
1146 Unclassified Traps
1147 DESCRIPTION
1147 Important Caveat
1147 Byte and Character semantics
1148 Effects of character semantics
1149 Character encodings for input and output
1149 CAVEATS
1149 SEE ALSO
1150 DESCRIPTION
1150 DOCUMENTATION
1150 CONVERTORS
1151 Development
1152 SEE ALSO
1153 DESCRIPTION
1153 Predefined Names
1165 Error Indicators
1166 Technical Note on the Syntax of Variable Names
1166 BUGS
1167 DESCRIPTION
1167 Introduction
1168 On The Road
1169 The Anatomy of an XSUB
1170 The Argument Stack
1170 The RETVAL Variable
1170 The MODULE Keyword
1171 The PACKAGE Keyword
1171 The PREFIX Keyword
1171 The OUTPUT: Keyword
1172 The NO_OUTPUT Keyword
1172 The CODE: Keyword
1173 The INIT: Keyword
1173 The NO_INIT Keyword
1174 Initializing Function Parameters
1175 Default Parameter Values
1175 The PREINIT: Keyword
1177 The SCOPE: Keyword
1177 The INPUT: Keyword
1178 The IN/OUTLIST/IN_OUTLIST/OUT/IN_OUT Keywords
1180 Variable-length Parameter Lists
1180 The C_ARGS: Keyword
1181 The PPCODE: Keyword
1182 Returning Undef And Empty Lists
1183 The REQUIRE: Keyword
1183 The CLEANUP: Keyword

1183 The POST_CALL: Keyword
1184 The BOOT: Keyword
1184 The VERSIONCHECK: Keyword
1184 The PROTOTYPES: Keyword
1184 The PROTOTYPE: Keyword
1185 The ALIAS: Keyword
1185 The INTERFACE: Keyword
1186 The INTERFACE_MACRO: Keyword
1186 The INCLUDE: Keyword
1187 The CASE: Keyword
1188 The & Unary Operator
1188 Inserting POD, Comments and C Preprocessor Directives
1189 Using XS With
1191 Interface Strategy
1191 Perl Objects And C Structures
1192 The Typemap
1193 EXAMPLES
1195 XS VERSION
1195 AUTHOR
1196 DESCRIPTION
1196 SPECIAL NOTES
1196 make
1196 Version caveat
1196 Dynamic Loading versus Static Loading
1197 TUTORIAL
1197 EXAMPLE 1
1199 EXAMPLE 2
1200 What has gone on?
1201 Writing good test scripts
1201 EXAMPLE 3
1202 What's new here?
1202 Input and Output Parameters
1202 The XSUBPP Program
1202 The TYPEMAP file
1203 Warning about Output Arguments
1203 EXAMPLE 4
1205 What has happened here?
1206 Anatomy of .xs file
1206 Getting the fat out of XSUBs
1207 More about XSUB arguments
1208 The Argument Stack
1209 Extending your Extension
1209 Documenting your Extension
1209 Installing your Extension
1210 EXAMPLE 5
1210 New Things in this Example
1211 EXAMPLE 6
1212 New Things in this Example
1213 EXAMPLE 7 (Coming Soon)
1213 EXAMPLE 8 (Coming Soon)
1213 EXAMPLE 9 (Coming Soon)
1213 Troubleshooting these Examples
1214 See also
1214 Author
1214 Last Changed

1215 SYNOPSIS
1215 DESCRIPTION
1215 Options
1215
1216 ENVIRONMENT
1216 AUTHOR
1216 FILES
1216 SEE ALSO
1216 DIAGNOSTICS
1216 BUGS
1217 SYNOPSIS
1217 OPTIONS
1217 DESCRIPTION
1221 SYNOPSIS
1221 DESCRIPTION
1221 OUTPUT
1222 OPTIONS
1223 ENVIRONMENT
1223 BUGS
1223 FILES
1223 SEE ALSO
1224 SYNOPSIS
1224 DESCRIPTION
1224 OPTIONS
1224 ENVIRONMENT
1224 FILES
1224 AUTHOR
1225 SEE ALSO
1225 DIAGNOSTICS
1225 BUGS
1226 SYNOPSIS
1226 DESCRIPTION
1226 OPTIONS
1227 EXAMPLES
1228 Extension based on .h and .c files
1229 ENVIRONMENT
1229 AUTHOR
1229 SEE ALSO
1229 DIAGNOSTICS
1229 LIMITATIONS of -x
1231 SYNOPSIS
1231 DESCRIPTION
1232 OPTIONS
1233 AUTHORS
1233 SEE ALSO
1233 BUGS
1234 SYNOPSIS
1234 DESCRIPTION
1234 OPTIONS
1235 ENVIRONMENT
1235 VERSION
1235 AUTHOR
1236 SYNOPSIS
1236 DESCRIPTION
1236 LIMITATIONS

1236 AUTHOR
1237 SYNOPSIS
1237 DESCRIPTION
1237 ARGUMENTS
1238 AUTHOR
1238 BUGS
1238 SEE ALSO
1238 COPYRIGHT
1239 SYNOPSIS
1239 DESCRIPTION
1239 OPTIONS
1240 DIAGNOSTICS
1240 EXAMPLES
1240 BUGS
1241 NOTES
1243 SEE ALSO
1243 AUTHOR
1245 SYNOPSIS
1245 DESCRIPTION
1245 Options
1245 Considerations
1245 ENVIRONMENT
1245 AUTHOR
1245 FILES
1245 SEE ALSO
1245 DIAGNOSTICS
1245 BUGS
1245 SYNOPSIS
1246 DESCRIPTION
1246 The diagnostics Pragma
1246 The splain Program
1246 EXAMPLES
1247 INTERNALS
1247 BUGS
1247 AUTHOR
1248 SYNOPSIS
1248 DESCRIPTION
1248 OPTIONS
1248 ENVIRONMENT
1248 AUTHOR
1249 MODIFICATION HISTORY
1249 SEE ALSO
1250 SYNOPSIS
1250 DESCRIPTION
1251 SYNOPSIS
1251 DESCRIPTION
1251 WARNING
1251 AUTHOR
1251 SEE ALSO
1252 SYNOPSIS
1252 DESCRIPTION
1252 HISTORY
1252 SEE ALSO
1253 SYNOPSIS
1253 DESCRIPTION

1253 BUGS
1253 AUTHOR
1254 SYNOPSIS
1254 DESCRIPTION
1254 NOTES
1255 TECHNICAL NOTE
1255 BUGS
1256 AUTHOR
1256 COPYRIGHT
1257 SYNOPSIS
1257 DESCRIPTION
1257 The diagnostics Pragma
1257 The splain Program
1257 EXAMPLES
1258 INTERNALS
1259 BUGS
1259 AUTHOR
1260 SYNOPSIS
1260 DESCRIPTION
1261 SEE ALSO
1262 SYNOPSIS
1262 DESCRIPTION
1262 subpragma access
1263 SYNOPSIS
1263 DESCRIPTION
1265 SYNOPSIS
1265 DESCRIPTION
1266 SYNOPSIS
1266 DESCRIPTION
1266 Adding directories to @INC
1266 Deleting directories from @INC
1266 Restoring original @INC
1266 SEE ALSO
1266 AUTHOR
1267 SYNOPSIS
1267 DESCRIPTION
1268 SYNOPSIS
1268 DESCRIPTION
1268 SEE ALSO
1269 SYNOPSIS
1269 DESCRIPTION
1269 Declaration of overloaded functions
1269 Calling Conventions for Binary Operations
1270 Calling Conventions for Unary Operations
1270 Calling Conventions for Mutators
1270 Overloadable Operations
1272 Inheritance and overloading
1272 SPECIAL SYMBOLS FOR use overload
1272 Last Resort
1273 Fallback
1273 Copy Constructor
1274 MAGIC AUTOGENERATION
1274 Losing overloading
1275 Run-time Overloading
1275 Public functions

1275 Overloading constants
1276 IMPLEMENTATION
1276 Metaphor clash
1277 Cookbook
1277 Two-face scalars
1278 Two-face references
1280 Symbolic calculator
1284 Really symbolic calculator
1285 AUTHOR
1285 DIAGNOSTICS
1285 BUGS
1287 SYNOPSIS
1287 DESCRIPTION
1289 SYNOPSIS
1289 DESCRIPTION
1289 OPTIONS
1289 SIGNAL HANDLERS
1289 SIGNAL LISTS
1290 OTHER
1290 EXAMPLES
1291 SYNOPSIS
1291 DESCRIPTION
1293 SYNOPSIS
1293 DESCRIPTION
1294 SYNOPSIS
1294 DESCRIPTION
1295 SYNOPSIS
1295 DESCRIPTION
1295 DBM Comparisons
1295 SEE ALSO
1296 SYNOPSIS
1296 DESCRIPTION
1296 Subroutine Stubs
1296 Using AutoLoader's AUTOLOAD Subroutine
1297 Overriding AutoLoader's AUTOLOAD Subroutine
1297 Package Lexicals
1297 Not Using AutoLoader
1297 AutoLoader vs. SelfLoader
1298 CAVEATS
1298 SEE ALSO
1299 SYNOPSIS
1299 DESCRIPTION
1299 Multiple packages
1300 DIAGNOSTICS
1301 SYNOPSIS
1301 DESCRIPTION
1301 OVERVIEW OF CLASSES
1301 SV-RELATED CLASSES
1301 B::SV METHODS
1301 B::IV METHODS
1301 B::NV METHODS
1302 B::RV METHODS
1302 B::PV METHODS
1302 B::PVMG METHODS
1302 B::MAGIC METHODS

1302 B::PVLV METHODS
1302 B::BM METHODS
1302 B::GV METHODS
1303 B::IO METHODS
1303 B::AV METHODS
1303 B::CV METHODS
1304 B::HV METHODS
1304 OP-RELATED CLASSES
1304 B::OP METHODS
1304 B::UNOP METHOD
1304 B::BINOP METHOD
1304 B::LOGOP METHOD
1304 B::LISTOP METHOD
1304 B::PMOP METHODS
1305 B::SVOP METHOD
1305 B::PADOP METHOD
1305 B::PVOP METHOD
1305 B::LOOP METHODS
1305 B::COP METHODS
1305 FUNCTIONS EXPORTED BY B
1306 AUTHOR
1307 SYNOPSIS
1307 DESCRIPTION
1307 AUTHOR
1308 SYNOPSIS
1308 DESCRIPTION
1308 AUTHORS
1309 SYNOPSIS
1309 DESCRIPTION
1309 AUTHOR
1310 SYNOPSIS
1310 DESCRIPTION
1310 OPTIONS
1311 EXAMPLES
1311 BUGS
1311 AUTHORS
1312 SYNOPSIS
1312 DESCRIPTION
1312 OPTIONS
1313 EXAMPLES
1313 BUGS
1313 AUTHOR
1314 SYNOPSIS
1314 DESCRIPTION
1314 OPTIONS
1315 EXAMPLES
1315 BUGS
1315 DIFFERENCES
1315 Loops
1315 Context of
1316 Arithmetic
1316 Deprecated features
1316 AUTHOR
1317 SYNOPSIS
1317 DESCRIPTION

1317 AUTHOR
1318 SYNOPSIS
1318 DESCRIPTION
1318 OPTIONS
1320 USING B::Deparse AS A MODULE
1320 Synopsis
1320 Description
1320 new
1321 coderef2text
1321 BUGS
1321 AUTHOR
1322 SYNOPSIS
1322 DESCRIPTION
1322 AUTHOR
1323 SYNOPSIS
1323 DESCRIPTION
1323 OPTIONS AND LINT CHECKS
1324 NON LINT-CHECK OPTIONS
1324 BUGS
1324 AUTHOR
1325 SYNOPSIS
1325 DESCRIPTION
1325 AUTHOR
1326 SYNOPSIS
1326 DESCRIPTION
1326 AUTHOR
1327 SYNOPSIS
1327 DESCRIPTION
1327 AUTHOR
1328 SYNOPSIS
1328 DESCRIPTION
1328 OPTIONS
1328 BUGS
1328 AUTHOR
1329 SYNOPSIS
1329 DESCRIPTION
1330 Methods
1330 Standard Exports
1331 Optional Exports
1332 NOTES
1332 EXAMPLES
1333 INHERITANCE
1333 CAVEATS
1333 SEE ALSO
1333 AUTHORS
1333 MODIFICATION HISTORY
1334 SYNOPSIS
1334 DESCRIPTION
1334 Forcing a Stack Trace
1334 BUGS
1335 SYNOPSIS
1335 ABSTRACT
1335 DESCRIPTION
1335 PROGRAMMING STYLE
1336 CALLING CGI.PM ROUTINES

1337 CREATING A NEW QUERY OBJECT (OBJECT-ORIENTED STYLE):
1338 CREATING A NEW QUERY OBJECT FROM AN INPUT FILE
1338 FETCHING A LIST OF KEYWORDS FROM THE QUERY:
1339 FETCHING THE NAMES OF ALL THE PARAMETERS PASSED TO YOUR SCRIPT:
1339 FETCHING THE VALUE OR VALUES OF A SINGLE NAMED PARAMETER:
1339 SETTING THE VALUE OF A NAMED PARAMETER:
1339 APPENDING ADDITIONAL VALUES TO A NAMED PARAMETER:
1339 IMPORTING ALL PARAMETERS INTO A NAMESPACE:
1340 DELETING A PARAMETER COMPLETELY:
1340 DELETING ALL PARAMETERS:
1340 DIRECT ACCESS TO THE PARAMETER LIST:
1340 FETCHING THE PARAMETER LIST AS A HASH:
1341 SAVING THE STATE OF THE SCRIPT TO A FILE:
1341 RETRIEVING CGI ERRORS
1342 USING THE FUNCTION-ORIENTED INTERFACE
1343 PRAGMAS
1345 SPECIAL FORMS FOR IMPORTING HTML-TAG FUNCTIONS
1346 GENERATING DYNAMIC DOCUMENTS
1346 CREATING A STANDARD HTTP HEADER:
1347 GENERATING A REDIRECTION HEADER
1348 CREATING THE HTML DOCUMENT HEADER
1351 ENDING THE HTML DOCUMENT:
1351 CREATING A SELF-REFERENCING URL THAT PRESERVES STATE INFORMATION:
1351 OBTAINING THE SCRIPT'S URL
1352 MIXING POST AND URL PARAMETERS
1352 CREATING STANDARD HTML ELEMENTS:
1353 PROVIDING ARGUMENTS TO HTML SHORTCUTS
1353 THE DISTRIBUTIVE PROPERTY OF HTML SHORTCUTS
1354 HTML SHORTCUTS AND LIST INTERPOLATION
1354 NON-STANDARD HTML SHORTCUTS
1355 AUTOESCAPING HTML
1355 PRETTY-PRINTING HTML
1355 CREATING FILL-OUT FORMS:
1356 CREATING AN ISINDEX TAG
1356 STARTING AND ENDING A FORM
1357 CREATING A TEXT FIELD
1358 CREATING A BIG TEXT FIELD
1358 CREATING A PASSWORD FIELD
1358 CREATING A FILE UPLOAD FIELD
1360 CREATING A POPUP MENU
1361 CREATING A SCROLLING LIST
1362 CREATING A GROUP OF RELATED CHECKBOXES
1363 CREATING A STANDALONE CHECKBOX
1363 CREATING A RADIO BUTTON GROUP
1364 CREATING A SUBMIT BUTTON
1365 CREATING A RESET BUTTON
1365 CREATING A DEFAULT BUTTON
1365 CREATING A HIDDEN FIELD
1366 CREATING A CLICKABLE IMAGE BUTTON
1366 CREATING A JAVASCRIPT ACTION BUTTON
1366 HTTP COOKIES
1368 WORKING WITH FRAMES
1369 LIMITED SUPPORT FOR CASCADING STYLE SHEETS
1370 DEBUGGING
1371 DUMPING OUT ALL THE NAME/VALUE PAIRS

1371 FETCHING ENVIRONMENT VARIABLES
1373 USING NPH SCRIPTS
1373 Server Push
1374 Avoiding Denial of Service Attacks
1375 COMPATIBILITY WITH CGI-LIB.PL
1376 AUTHOR INFORMATION
1376 CREDITS
1377 A COMPLETE EXAMPLE OF A SIMPLE FORM-BASED SCRIPT
1378 BUGS
1378 SEE ALSO
1379 SYNOPSIS
1379 ABSTRACT
1379 DESCRIPTION
1379 AUTHOR INFORMATION
1379 BUGS
1379 SEE ALSO
1379 SYNOPSIS
1379 DESCRIPTION
1379 REDIRECTING ERROR MESSAGES
1380 MAKING PERL ERRORS APPEAR IN THE BROWSER WINDOW
1380 Changing the default message
1381 MAKING WARNINGS APPEAR AS HTML COMMENTS
1381 CHANGE LOG
1382 AUTHORS
1382 SEE ALSO
1383 SYNOPSIS
1383 DESCRIPTION
1383 USING CGI::Cookie
1384 Creating New Cookies
1384 Sending the Cookie to the Browser
1385 Recovering Previous Cookies
1385 Manipulating Cookies
1385 AUTHOR INFORMATION
1386 BUGS
1386 SEE ALSO
1387 SYNOPSIS
1387 DESCRIPTION
1387 OTHER PIECES OF THE PUZZLE
1387 WRITING FASTCGI PERL SCRIPTS
1388 INSTALLING FASTCGI SCRIPTS
1388 USING FASTCGI SCRIPTS AS CGI SCRIPTS
1388 CAVEATS
1388 AUTHOR INFORMATION
1388 BUGS
1388 SEE ALSO
1389 SYNOPSIS
1389 DESCRIPTION
1389 USING CGI::Push
1390 Heterogeneous Pages
1391 Changing the Page Delay on the Fly
1391 INSTALLING CGI::Push SCRIPTS
1391 AUTHOR INFORMATION
1392 BUGS
1392 SEE ALSO
1393 SYNOPSIS

Perl Table of Contents

1393 ABSTRACT
1393 DESCRIPTION
1393 AUTHOR INFORMATION
1393 BUGS
1393 SEE ALSO
1393 SYNOPSIS
1394 DESCRIPTION
1394 The struct() function
1394 Class Creation at Compile Time
1394 Element Types and Accessor Methods
1395 Initializing with new
1395 EXAMPLES
1397 Author and Modification History
1399 SYNOPSIS
1399 DESCRIPTION
1399 EXAMPLE
1400 WARNING
1400 GLOSSARY
1400 _
1400 a
1402 b
1402 c
1403 C
1404 d
1420 e
1420 f
1421 g
1421 h
1421 i
1427 k
1427 l
1429 m
1429 M
1430 n
1431 o
1432 p
1432 P
1433 q
1433 r
1434 s
1438 t
1438 u
1440 v
1441 x
1441 z
1441 NOTE
1442 SYNOPSIS
1442 DESCRIPTION
1442 Interactive Mode
1444 CPAN::Shell
1444 autobundle
1444 recompile
1444 The four CPAN::* Classes: Author, Bundle, Module, Distribution
1445 Programmer's interface
1446 Methods in the other Classes

Perl Table of Contents

1449 Cache Manager
1449 Bundles
1450 Prerequisites
1450 Finding packages and VERSION
1450 Debugging
1451 Floppy, Zip, Offline Mode
1451 CONFIGURATION
1452 Note on urllist parameter's format
1452 urllist parameter has CD-ROM support
1452 SECURITY
1452 EXPORT
1452 POPULATE AN INSTALLATION WITH LOTS OF MODULES
1452 WORKING WITH CPAN.pm BEHIND FIREWALLS
1453 Three basic types of firewalls
1453 Configuring lynx or ncftp for going through a firewall
1453 FAQ
1455 BUGS
1455 AUTHOR
1455 TRANSLATIONS
1455 SEE ALSO
1456 SYNOPSIS
1456 DESCRIPTION
1457 SYNOPSIS
1457 DESCRIPTION
1457 SEE ALSO
1458 SYNOPSIS
1458 DESCRIPTION
1459 SYNOPSIS
1459 DESCRIPTION
1460 Methods
1460 Functions
1460 Configuration Variables or Methods
1462 Exports
1462 EXAMPLES
1464 BUGS
1465 AUTHOR
1465 VERSION
1465 SEE ALSO
1466 SYNOPSIS
1466 DESCRIPTION
1467 SYNOPSIS
1467 DESCRIPTION
1468 SYNOPSIS
1468 DESCRIPTION
1473 AUTHOR
1474 SYNOPSIS
1474 DESCRIPTION
1474 Creation
1474 Methods
1476 SYNOPSIS
1476 DESCRIPTION
1476 BUGS
1477 SYNOPSIS
1477 DESCRIPTION
1477 LIMITATIONS

Perl Table of Contents

1477 AUTHOR
1478 SYNOPSIS
1478 DESCRIPTION
1478 CAVEATS
1478 AUTHOR
1478 COPYRIGHT
1479 SYNOPSIS
1479 DESCRIPTION
1479 How to Export
1479 Selecting What To Export
1480 Specialised Import Lists
1480 Exporting without using Export's import method
1481 Module Version Checking
1481 Managing Unknown Symbols
1481 Tag Handling Utility Functions
1483 SYNOPSIS
1483 DESCRIPTION
1483 BUGS
1483 SEE ALSO
1483 AUTHOR
1484 SYNOPSIS
1484 DESCRIPTION
1484 @EXPORT
1484 FUNCTIONS
1486 EXAMPLES
1486 SEE ALSO
1486 AUTHOR
1487 SYNOPSIS
1487 DESCRIPTION
1488 SYNOPSIS
1488 DESCRIPTION
1488 USAGE
1488 FUNCTIONS
1489 EXAMPLE
1489 AUTHOR
1490 SYNOPSIS
1490 DESCRIPTION
1490 EXTRALIBS
1490 LDLOADLIBS and LD_RUN_PATH
1490 BSLOADLIBS
1490 PORTABILITY
1490 VMS implementation
1491 Win32 implementation
1492 SEE ALSO
1493 SYNOPSIS
1493 DESCRIPTION
1493 How To Write A Makefile.PL
1493 Default Makefile Behaviour
1494 make test
1494 make testdb
1494 make install
1495 PREFIX and LIB attribute
1495 AFS users
1495 Static Linking of a new Perl Binary
1496 Determination of Perl Library and Installation Locations

1497 Which architecture dependent directory?
1497 Using Attributes and Parameters
1505 Additional lowercase attributes
1506 Overriding MakeMaker Methods
1506 Hintsfile support
1507 Distribution Support
1508 Disabling an extension
1508 ENVIRONMENT
1508 SEE ALSO
1508 AUTHORS
1509 SYNOPSIS
1509 DESCRIPTION
1510 MANIFEST.SKIP
1510 EXPORT_OK
1510 GLOBAL VARIABLES
1510 DIAGNOSTICS
1510 SEE ALSO
1510 AUTHOR
1511 SYNOPSIS
1511 DESCRIPTION
1511 SEE ALSO
1512 SYNOPSIS
1512 DESCRIPTION
1513 SYNOPSIS
1513 DESCRIPTION
1514 AUTHOR
1514 REVISION
1515 SYNOPSIS
1515 DESCRIPTION
1516 SYNOPSIS
1516 DESCRIPTION
1516 METHODS
1516 Preloaded methods
1516 SelfLoaded methods
1518 Methods to actually produce chunks of text for the Makefile
1521 SEE ALSO
1522 SYNOPSIS
1522 DESCRIPTION
1522 Methods always loaded
1522 SelfLoaded methods
1525 SYNOPSIS
1525 DESCRIPTION
1527 SYNOPSIS
1527 DESCRIPTION
1527 USAGE
1527 FUNCTIONS
1527 EXAMPLE
1528 AUTHOR
1529 SYNOPSIS
1529 DESCRIPTION
1530 SYNOPSIS
1530 DESCRIPTION
1530 AUTHOR
1531 SYNOPSIS
1531 DESCRIPTION

1531 NOTE
1531 EXPORTED SYMBOLS
1532 SYNOPSIS
1532 DESCRIPTION
1532 EXAMPLES
1534 SYNOPSIS
1534 DESCRIPTION
1535 SYNOPSIS
1535 DESCRIPTION
1535 RETURN
1535 AUTHOR
1536 SYNOPSIS
1536 DESCRIPTION
1536 Special behaviour if syscopy is defined (OS/2, VMS and Win32)
1537 RETURN
1537 AUTHOR
1538 SYNOPSIS
1538 DESCRIPTION
1538 EXPORTS (by request only)
1538 BUGS
1538 AUTHOR
1538 HISTORY
1539 SEE ALSO
1540 SYNOPSIS
1540 DESCRIPTION
1541 CAVEAT
1542 SYNOPSIS
1542 DESCRIPTION
1542 AUTHORS
1543 SYNOPSIS
1543 DESCRIPTION
1543 SEE ALSO
1543 AUTHORS
1545 SYNOPSIS
1545 DESCRIPTION
1545 METHODS
1546 SEE ALSO
1547 SYNOPSIS
1547 DESCRIPTION
1548 SYNOPSIS
1548 DESCRIPTION
1548 METHODS
1550 SEE ALSO
1551 SYNOPSIS
1551 DESCRIPTION
1551 Methods always loaded
1552 SEE ALSO
1553 SYNOPSIS
1553 DESCRIPTION
1554 SEE ALSO
1555 SYNOPSIS
1555 DESCRIPTION
1555 NOTE
1555 AUTHOR
1556 SYNOPSIS

Perl Table of Contents

1556 DESCRIPTION
1556 BUGS
1557 SYNOPSIS
1557 DESCRIPTION
1559 SEE ALSO
1560 SYNOPSIS
1560 DESCRIPTION
1560 EXPORTABLE VARIABLES
1560 KNOWN BUGS
1560 AUTHORS
1560 COPYRIGHT
1561 SYNOPSIS
1561 DESCRIPTION
1561 Command Line Options, an Introduction
1561 Getting Started with Getopt::Long
1562 Simple options
1562 A little bit less simple options
1562 Mixing command line option with other arguments
1563 Options with values
1563 Options with multiple values
1563 Options with hash values
1564 User-defined subroutines to handle options
1564 Options with multiple names
1564 Case and abbreviations
1564 Summary of Option Specifications
1565 Advanced Possibilities
1565 Object oriented interface
1565 Documentation and help texts
1566 Storing options in a hash
1567 Bundling
1568 The lonesome dash
1568 Argument call-back
1568 Configuring Getopt::Long
1570 Return values and Errors
1570 Legacy
1571 Default destinations
1571 Alternative option starters
1571 Configuration variables
1571 Trouble Shooting
1571 Warning: Ignoring '!' modifier for short option
1572 GetOptions does not return a false result when an option is not supplied
1572 AUTHOR
1572 COPYRIGHT AND DISCLAIMER
1573 SYNOPSIS
1573 DESCRIPTION
1574 SYNOPSIS
1574 DESCRIPTION
1575 SYNOPSIS
1575 DESCRIPTION
1576 SYNOPSIS
1576 DESCRIPTION
1576 CONSTRUCTOR
1576 METHODS
1577 SEE ALSO
1577 HISTORY

1578 SYNOPSIS
1578 DESCRIPTION
1578 CONSTRUCTOR
1578 METHODS
1580 NOTE
1580 SEE ALSO
1580 BUGS
1580 HISTORY
1581 SYNOPSIS
1581 DESCRIPTION
1581 CONSTRUCTOR
1581 METHODS
1582 SEE ALSO
1582 AUTHOR
1582 COPYRIGHT
1583 SYNOPSIS
1583 DESCRIPTION
1583 SEE ALSO
1583 HISTORY
1584 SYNOPSIS
1584 DESCRIPTION
1584 CONSTRUCTOR
1584 METHODS
1585 EXAMPLE
1585 AUTHOR
1585 COPYRIGHT
1586 SYNOPSIS
1586 DESCRIPTION
1586 CONSTRUCTOR
1586 METHODS
1587 SEE ALSO
1587 AUTHOR
1587 COPYRIGHT
1588 SYNOPSIS
1588 DESCRIPTION
1588 METHODS
1589 SEE ALSO
1589 AUTHOR
1589 COPYRIGHT
1590 SYNOPSIS
1590 DESCRIPTION
1590 WARNING
1591 SEE ALSO
1592 SYNOPSIS
1592 DESCRIPTION
1592 WARNING
1593 SYNOPSIS
1593 DESCRIPTION
1593 METHODS
1594 SEE ALSO
1594 AUTHOR
1594 COPYRIGHT
1595 SYNOPSIS
1595 DESCRIPTION
1595 SEE ALSO

1595 AUTHORS
1595 COPYRIGHT
1596 SYNOPSIS
1596 DESCRIPTION
1596 BUGS
1597 AUTHOR
1598 SYNOPSIS
1598 DESCRIPTION
1598 EXAMPLES
1598 Autocreating constants
1599 BUGS
1599 AUTHOR
1600 SYNOPSIS
1600 DESCRIPTION
1602 OPERATIONS
1603 CREATION
1604 STRINGIFICATION
1604 CHANGED IN PERL 5.6
1605 USAGE
1605 ERRORS DUE TO DIVISION BY ZERO OR LOGARITHM OF ZERO
1606 ERRORS DUE TO INDIGESTIBLE ARGUMENTS
1606 BUGS
1606 AUTHORS
1607 SYNOPSIS
1607 DESCRIPTION
1607 TRIGONOMETRIC FUNCTIONS
1608 ERRORS DUE TO DIVISION BY ZERO
1608 SIMPLE (REAL) ARGUMENTS, COMPLEX RESULTS
1609 PLANE ANGLE CONVERSIONS
1609 RADIAL COORDINATE CONVERSIONS
1609 COORDINATE SYSTEMS
1610 3-D ANGLE CONVERSIONS
1610 GREAT CIRCLE DISTANCES
1610 EXAMPLES
1611 BUGS
1611 AUTHORS
1612 SYNOPSIS
1612 DESCRIPTION
1612 DIAGNOSTICS
1612 ndbm store returned -1, errno 22, key "...
1612 BUGS AND WARNINGS
1613 SYNOPSIS
1613 DESCRIPTION
1613 EXAMPLES
1614 NOTE
1614 AUTHOR
1615 SYNOPSIS
1615 DESCRIPTION
1615 EXAMPLES
1616 NOTE
1616 AUTHOR
1617 SYNOPSIS
1617 DESCRIPTION
1618 Functions
1618 WARNING

1618 NOTES
1619 SYNOPSIS
1619 DESCRIPTION
1619 NOTE
1619 AUTHOR
1620 SYNOPSIS
1620 DESCRIPTION
1620 EXAMPLES
1620 NOTE
1620 AUTHOR
1621 SYNOPSIS
1621 DESCRIPTION
1621 CONVENTIONS
1621 IMPLEMENTATION
1621 AUTHOR
1622 SYNOPSIS
1622 DESCRIPTION
1622 NOTE
1622 WARNING
1622 Operator Names and Operator Lists
1623 Opcode Functions
1624 Manipulating Opsets
1624 TO DO (maybe)
1624 Predefined Opcode Tags
1628 SEE ALSO
1628 AUTHORS
1629 SYNOPSIS
1629 DESCRIPTION
1629 ARGUMENTS
1631 EXAMPLE
1631 ENVIRONMENT
1631 AUTHOR
1631 SEE ALSO
1631 COPYRIGHT
1632 SYNOPSIS
1632 DESCRIPTION
1633 DIAGNOSTICS
1633 RESTRICTIONS
1633 NOTES
1633 SEE ALSO
1633 AUTHOR
1634 SYNOPSIS
1634 DESCRIPTION
1634 NOTE
1634 CAVEATS
1634 FUNCTIONS
1634 _exit
1634 abort
1634 abs
1635 access
1635 acos
1635 alarm
1635 asctime
1635 asin
1635 assert

1635 atan
1635 atan2
1635 atexit
1635 atof
1635 atoi
1635 atol
1635 bsearch
1635 calloc
1635 ceil
1636 chdir
1636 chmod
1636 chown
1636 clearerr
1636 clock
1636 close
1636 closedir
1636 cos
1636 cosh
1636 creat
1636 ctermid
1636 ctime
1636 cuserid
1636 difftime
1636 div
1636 dup
1637 dup2
1637 errno
1637 execl
1637 execle
1637 execlp
1637 execv
1637 execve
1637 execvp
1637 exit
1637 exp
1637 fabs
1637 fclose
1637 fcntl
1637 fdopen
1637 feof
1637 ferror
1637 fflush
1637 fgetc
1637 fgetpos
1637 fgets
1637 fileno
1637 floor
1637 fmod
1638 fopen
1638 fork
1638 fpathconf
1638 fprintf
1638 fputc
1638 fputs
1638 fread

1638 free
1638 freopen
1638 frexp
1638 fscanf
1638 fseek
1638 fsetpos
1638 fstat
1638 ftell
1638 fwrite
1638 getc
1638 getchar
1638 getcwd
1638 getegid
1638 getenv
1638 geteuid
1639 getgid
1639 getgrgid
1639 getgrnam
1639 getgroups
1639 getlogin
1639 getpgrp
1639 getpid
1639 getppid
1639 getpwnam
1639 getpwuid
1639 gets
1639 getuid
1639 gmtime
1639 isalnum
1639 isalpha
1639 isatty
1639 iscntrl
1639 isdigit
1640 isgraph
1640 islower
1640 isprint
1640 ispunct
1640 isspace
1640 isupper
1640 isxdigit
1640 kill
1640 labs
1640 ldexp
1640 ldiv
1640 link
1640 localeconv
1641 localtime
1641 log
1641 log10
1641 longjmp
1641 lseek
1641 malloc
1641 mblen
1642 mbstowcs
1642 mbtowc

1642 memchr
1642 memcmp
1642 memcpy
1642 memmove
1642 memset
1642 mkdir
1642 mkfifo
1642 mktime
1642 modf
1642 nice
1642 offsetof
1643 open
1643 opendir
1643 pathconf
1643 pause
1643 perror
1643 pipe
1643 pow
1644 printf
1644 putc
1644 putchar
1644 puts
1644 qsort
1644 raise
1644 rand
1644 read
1644 readdir
1644 realloc
1644 remove
1644 rename
1644 rewind
1644 rewinddir
1644 rmdir
1644 scanf
1644 setgid
1644 setjmp
1644 setlocale
1645 setpgid
1645 setsid
1645 setuid
1645 sigaction
1645 siglongjmp
1645 sigpending
1645 sigprocmask
1645 sigsetjmp
1646 sigsuspend
1646 sin
1646 sinh
1646 sleep
1646 sprintf
1646 sqrt
1646 srand
1646 sscanf
1646 stat
1646 strcat

1646 strchr
1646 strcmp
1646 strcoll
1646 strcpy
1646 strcspn
1646 strerror
1646 strftime
1647 strlen
1647 strncat
1647 strncmp
1647 strncpy
1647 strpbrk
1647 strrchr
1647 strspn
1647 strstr
1647 strtod
1647 strtok
1647 strtol
1648 strtoul
1648 strxfrm
1648 sysconf
1648 system
1648 tan
1648 tanh
1648 tcdrain
1648 tcflow
1648 tcflush
1648 tcgetpgrp
1648 tcsendbreak
1648 tcsetpgrp
1649 time
1649 times
1649 tmpfile
1649 tmpnam
1649 tolower
1649 toupper
1649 ttyname
1649 tzname
1649 tzset
1649 umask
1649 uname
1649 ungetc
1649 unlink
1649 utime
1650 vfprintf
1650 vprintf
1650 vsprintf
1650 wait
1650 waitpid
1650 wcstombs
1650 wctomb
1650 write
1650 CLASSES
1650 POSIX::SigAction
1650 new

1650 POSIX::SigSet
1650 new
1650 addset
1651 delset
1651 emptyset
1651 fillset
1651 ismember
1651 POSIX::Termios
1651 new
1651 getattr
1651 getcc
1651 getcflag
1652 getiflag
1652 getispeed
1652 getlflag
1652 getoflag
1652 getospeed
1652 setattr
1652 setcc
1652 setcflag
1652 setiflag
1652 setispeed
1652 setlflag
1652 setoflag
1653 setospeed
1653 Baud rate values
1653 Terminal interface values
1653 c_cc field values
1653 c_cflag field values
1653 c_iflag field values
1653 c_lflag field values
1653 c_oflag field values
1653 PATHNAME CONSTANTS
1653 Constants
1653 POSIX CONSTANTS
1653 Constants
1653 SYSTEM CONFIGURATION
1653 Constants
1653 ERRNO
1653 Constants
1654 FCNTL
1654 Constants
1654 FLOAT
1654 Constants
1654 LIMITS
1654 Constants
1654 LOCALE
1654 Constants
1654 MATH
1654 Constants
1654 SIGNAL
1654 Constants
1654 STAT
1654 Constants
1654 Macros

1654 STDLIB
1654 Constants
1654 STDIO
1654 Constants
1654 TIME
1654 Constants
1655 UNISTD
1655 Constants
1655 WAIT
1655 Constants
1655 Macros
1656 SYNOPSIS
1656 DESCRIPTION
1656 WARNING
1657 RECENT CHANGES
1657 Methods in class Safe
1659 Some Safety Issues
1659 AUTHOR
1660 SYNOPSIS
1660 DESCRIPTION
1660 DIAGNOSTICS
1660 sdbm store returned -1, errno 22, key "...
1660 BUGS AND WARNINGS
1661 SYNOPSIS
1661 DESCRIPTION
1662 SYNOPSIS
1662 DESCRIPTION
1663 SYNOPSIS
1663 DESCRIPTION
1663 The __DATA__ token
1663 SelfLoader autoloading
1663 Autoloading and package lexicals
1664 SelfLoader and AutoLoader
1664 __DATA__, __END__, and the FOOBAR::DATA filehandle.
1664 Classes and inherited methods.
1665 Multiple packages and fully qualified subroutine names
1666 SYNOPSIS
1666 DESCRIPTION
1667 OBJECT ORIENTED SYNTAX
1667 AUTHOR
1668 SYNOPSIS
1668 DESCRIPTION
1670 SYNOPSIS
1670 DESCRIPTION
1671 SYNOPSIS
1671 DESCRIPTION
1671 AUTHOR
1672 SYNOPSIS
1672 DESCRIPTION
1672 EXAMPLES
1673 SEE ALSO
1673 AUTHOR
1674 SYNOPSIS
1674 DESCRIPTION
1674 EXAMPLES

1676 SYNOPSIS
1676 DESCRIPTION
1676 DIAGNOSTICS
1676 BUGS
1676 AUTHOR
1677 SYNOPSIS
1677 DESCRIPTION
1677 Minimal set of supported functions
1677 Additional supported functions
1678 EXPORTS
1678 ENVIRONMENT
1679 SYNOPSIS
1679 DESCRIPTION
1679 TEST TYPES
1680 RETURN VALUE
1680 ONFAIL
1680 SEE ALSO
1680 AUTHOR
1681 SYNOPSIS
1681 DESCRIPTION
1681 The test script output
1681 EXPORT
1682 DIAGNOSTICS
1682 ENVIRONMENT
1682 SEE ALSO
1682 AUTHORS
1682 BUGS
1683 SYNOPSIS
1683 DESCRIPTION
1683 EXAMPLE
1684 SYNOPSIS
1684 DESCRIPTION
1684 EXAMPLES
1685 AUTHORS
1686 SYNOPSIS
1686 DESCRIPTION
1686 EXAMPLES
1686 LIMITATIONS
1686 AUTHOR
1688 SYNOPSIS
1688 DESCRIPTION
1688 BUGS
1688 AUTHOR
1689 SYNOPSIS
1689 DESCRIPTION
1690 EXAMPLE
1690 AUTHOR
1691 SYNOPSIS
1691 DESCRIPTION
1693 CAVEATS
1693 AUTHOR
1694 SYNOPSIS
1694 DESCRIPTION
1695 MORE INFORMATION
1695 COMPATIBILITY

Perl Table of Contents

1696 SYNOPSIS
1696 DESCRIPTION
1697 CAVEATS
1697 MORE INFORMATION
1698 SYNOPSIS
1698 DESCRIPTION
1698 EXAMPLE
1698 AUTHOR
1698 VERSION
1698 SEE ALSO
1699 SYNOPSIS
1699 DESCRIPTION
1699 MORE INFORMATION
1700 SYNOPSIS
1700 DESCRIPTION
1700 CAVEATS
1701 SYNOPSIS
1701 DESCRIPTION
1701 NOTE
1701 AUTHOR
1702 SYNOPSIS
1702 DESCRIPTION
1703 IMPLEMENTATION
1703 BUGS
1704 SYNOPSIS
1704 DESCRIPTION
1704 NOTE
1704 AUTHOR
1705 SYNOPSIS
1705 DESCRIPTION
1705 AUTHOR
1706 SYNOPSIS
1706 DESCRIPTION
1707 SYNOPSIS
1707 DESCRIPTION
1707 NOTE
1707 AUTHOR
1708 SYNOPSIS
1708 DESCRIPTION
1709 System Specifics
1709 NOTE
1709 AUTHOR
1709 HISTORY