NAME

feature - Perl pragma to enable new features

SYNOPSIS

```
use feature qw(fc say);

# Without the "use feature" above, this code would not be able to find
# the built-ins "say" or "fc":
say "The case-folded version of $x is: " . fc $x;



# set features to match the :5.10 bundle, which may turn off or on
# multiple features (see below)
use feature ':5.10';



# implicitly loads :5.10 feature bundle
use v5.10;
```

DESCRIPTION

It is usually impossible to add new syntax to Perl without breaking some existing programs. This pragma provides a way to minimize that risk. New syntactic constructs, or new semantic meanings to older constructs, can be enabled by "use feature 'foo'", and will be parsed only when the appropriate feature pragma is in scope. (Nevertheless, the "CORE::" prefix provides access to all Perl keywords, regardless of this pragma.)

Lexical effect

Like other pragmas ("use strict", for example), features have a lexical effect. "use feature qw(foo)" will only make the feature "foo" available from that point to the end of the enclosing block.

```
{
   use feature 'say';
   say "say is available here";
}
print "But not here.\n";
```

"no feature"

Features can also be turned off by using "no feature "foo"". This too has lexical effect.

```
use feature 'say';
```

```
say "say is available here";
{
   no feature ’say’;
   print "But not here.\n";
}
say "Yet it is here.";
```

"no feature" with no features specified will reset to the default group. To disable *all* features (an unusual request!) use "no feature ’:all’".

AVAILABLE FEATURES

The ’say’ feature

"use feature ’say’" tells the compiler to enable the Raku-inspired "say" function.

See "say" in perlfunc for details.

This feature is available starting with Perl 5.10.

The ’state’ feature

"use feature ’state’" tells the compiler to enable "state" variables.

See "Persistent Private Variables" in perlsub for details.

This feature is available starting with Perl 5.10.

The ’switch’ feature

WARNING: This feature is still experimental and the implementation may change or be removed in future versions of Perl. For this reason, Perl will warn when you use the feature, unless you have explicitly disabled the warning:

```
no warnings "experimental::smartmatch";
```

"use feature ’switch’" tells the compiler to enable the Raku given/when construct.

See "Switch Statements" in perlsyn for details.

This feature is available starting with Perl 5.10.

The ’unicode_strings’ feature

"use feature ’unicode_strings’" tells the compiler to use Unicode rules in all string operations executed

within its scope (unless they are also within the scope of either "use locale" or "use bytes"). The same applies to all regular expressions compiled within the scope, even if executed outside it. It does not change the internal representation of strings, but only how they are interpreted.

"no feature 'unicode_strings'" tells the compiler to use the traditional Perl rules wherein the native character set rules is used unless it is clear to Perl that Unicode is desired. This can lead to some surprises when the behavior suddenly changes. (See "The "Unicode Bug"" in perlunicode for details.) For this reason, if you are potentially using Unicode in your program, the "use feature 'unicode_strings'" subpragma is **strongly** recommended.

This feature is available starting with Perl 5.12; was almost fully implemented in Perl 5.14; and extended in Perl 5.16 to cover "quotemeta"; was extended further in Perl 5.26 to cover the range operator; and was extended again in Perl 5.28 to cover special-cased whitespace splitting.

The 'unicode_eval' and 'evalbytes' features

Together, these two features are intended to replace the legacy string "eval" function, which behaves problematically in some instances. They are available starting with Perl 5.16, and are enabled by default by a "use 5.16" or higher declaration.

"unicode_eval" changes the behavior of plain string "eval" to work more consistently, especially in the Unicode world. Certain (mis)behaviors couldn't be changed without breaking some things that had come to rely on them, so the feature can be enabled and disabled. Details are at "Under the "unicode_eval" feature" in perlfunc.

"evalbytes" is like string "eval", but it treats its argument as a byte string. Details are at "evalbytes EXPR" in perlfunc. Without a "use feature 'evalbytes'" nor a "use v5.16" (or higher) declaration in the current scope, you can still access it by instead writing "CORE::evalbytes".

The 'current_sub' feature

This provides the "__SUB__" token that returns a reference to the current subroutine or "undef" outside of a subroutine.

This feature is available starting with Perl 5.16.

The 'array_base' feature

This feature supported the legacy $[variable. See "$[" in perlvar. It was on by default but disabled under "use v5.16" (see "IMPLICIT LOADING", below) and unavailable since perl 5.30.

This feature is available under this name starting with Perl 5.16. In previous versions, it was simply on all the time, and this pragma knew nothing about it.

The ’fc’ feature

"use feature ’fc’" tells the compiler to enable the "fc" function, which implements Unicode casefolding.

See "fc" in perlfunc for details.

This feature is available from Perl 5.16 onwards.

The ’lexical_subs’ feature

In Perl versions prior to 5.26, this feature enabled declaration of subroutines via "my sub foo", "state sub foo" and "our sub foo" syntax. See "Lexical Subroutines" in perlsub for details.

This feature is available from Perl 5.18 onwards. From Perl 5.18 to 5.24, it was classed as experimental, and Perl emitted a warning for its usage, except when explicitly disabled:

```
no warnings "experimental::lexical_subs";
```

As of Perl 5.26, use of this feature no longer triggers a warning, though the "experimental::lexical_subs" warning category still exists (for compatibility with code that disables it). In addition, this syntax is not only no longer experimental, but it is enabled for all Perl code, regardless of what feature declarations are in scope.

The ’postderef’ and ’postderef_qq’ features

The ’postderef_qq’ feature extends the applicability of postfix dereference syntax so that postfix array and scalar dereference are available in double-quotish interpolations. For example, it makes the following two statements equivalent:

```
my $s = "[@{ $h->{a} }]";
my $s = "[$h->{a}->@*]";
```

This feature is available from Perl 5.20 onwards. In Perl 5.20 and 5.22, it was classed as experimental, and Perl emitted a warning for its usage, except when explicitly disabled:

```
no warnings "experimental::postderef";
```

As of Perl 5.24, use of this feature no longer triggers a warning, though the "experimental::postderef" warning category still exists (for compatibility with code that disables it).

The ’postderef’ feature was used in Perl 5.20 and Perl 5.22 to enable postfix dereference syntax outside double-quotish interpolations. In those versions, using it triggered the "experimental::postderef" warning in the same way as the ’postderef_qq’ feature did. As of Perl 5.24, this syntax is not only no

longer experimental, but it is enabled for all Perl code, regardless of what feature declarations are in scope.

The 'signatures' feature

This enables syntax for declaring subroutine arguments as lexical variables. For example, for this subroutine:

```
sub foo ($left, $right) {
    return $left + $right;
}
```

Calling "foo(3, 7)" will assign 3 into $left and 7 into $right.

See "Signatures" in perlsub for details.

This feature is available from Perl 5.20 onwards. From Perl 5.20 to 5.34, it was classed as experimental, and Perl emitted a warning for its usage, except when explicitly disabled:

```
 no warnings "experimental::signatures";
```

As of Perl 5.36, use of this feature no longer triggers a warning, though the "experimental::signatures" warning category still exists (for compatibility with code that disables it). This feature is now considered stable, and is enabled automatically by "use v5.36" (or higher).

The 'refaliasing' feature

WARNING: This feature is still experimental and the implementation may change or be removed in future versions of Perl. For this reason, Perl will warn when you use the feature, unless you have explicitly disabled the warning:

```
no warnings "experimental::refaliasing";
```

This enables aliasing via assignment to references:

```
\$a = \$b; # $a and $b now point to the same scalar
\@a = \@b; #                    to the same array
\%a = \%b;
\&a = \&b;
foreach \%hash (@array_of_hash_refs) {
    ...
}
```

See "Assigning to References" in perlref for details.

This feature is available from Perl 5.22 onwards.

The 'bitwise' feature

This makes the four standard bitwise operators ("& | ^ ~") treat their operands consistently as numbers, and introduces four new dotted operators ("&. |. ^. ~.") that treat their operands consistently as strings. The same applies to the assignment variants ("&= |= ^= &.= |.= ^.=").

See "Bitwise String Operators" in perlop for details.

This feature is available from Perl 5.22 onwards. Starting in Perl 5.28, "use v5.28" will enable the feature. Before 5.28, it was still experimental and would emit a warning in the "experimental::bitwise" category.

The 'declared_refs' feature

WARNING: This feature is still experimental and the implementation may change or be removed in future versions of Perl. For this reason, Perl will warn when you use the feature, unless you have explicitly disabled the warning:

```
no warnings "experimental::declared_refs";
```

This allows a reference to a variable to be declared with "my", "state", our "our", or localized with "local". It is intended mainly for use in conjunction with the "refaliasing" feature. See "Declaring a Reference to a Variable" in perlref for examples.

This feature is available from Perl 5.26 onwards.

The 'isa' feature

This allows the use of the "isa" infix operator, which tests whether the scalar given by the left operand is an object of the class given by the right operand. See "Class Instance Operator" in perlop for more details.

This feature is available from Perl 5.32 onwards. From Perl 5.32 to 5.34, it was classed as experimental, and Perl emitted a warning for its usage, except when explicitly disabled:

```
no warnings "experimental::isa";
```

As of Perl 5.36, use of this feature no longer triggers a warning (though the "experimental::isa" warning category stilll exists for compatibility with code that disables it). This feature is now

considered stable, and is enabled automatically by "use v5.36" (or higher).

The 'indirect' feature

This feature allows the use of indirect object syntax for method calls, e.g. "new Foo 1, 2;". It is enabled by default, but can be turned off to disallow indirect object syntax.

This feature is available under this name from Perl 5.32 onwards. In previous versions, it was simply on all the time. To disallow (or warn on) indirect object syntax on older Perls, see the indirect CPAN module.

The 'multidimensional' feature

This feature enables multidimensional array emulation, a perl 4 (or earlier) feature that was used to emulate multidimensional arrays with hashes. This works by converting code like $foo{$x, $y} into $foo{join($;, $x, $y)}. It is enabled by default, but can be turned off to disable multidimensional array emulation.

When this feature is disabled the syntax that is normally replaced will report a compilation error.

This feature is available under this name from Perl 5.34 onwards. In previous versions, it was simply on all the time.

You can use the multidimensional module on CPAN to disable multidimensional array emulation for older versions of Perl.

The 'bareword_filehandles' feature.

This feature enables bareword filehandles for builtin functions operations, a generally discouraged practice. It is enabled by default, but can be turned off to disable bareword filehandles, except for the exceptions listed below.

The perl built-in filehandles "STDIN", "STDOUT", "STDERR", "DATA", "ARGV", "ARGVOUT" and the special "_" are always enabled.

This feature is enabled under this name from Perl 5.34 onwards. In previous versions it was simply on all the time.

You can use the bareword::filehandles module on CPAN to disable bareword filehandles for older versions of perl.

The 'try' feature.

WARNING: This feature is still experimental and the implementation may change or be removed in

future versions of Perl. For this reason, Perl will warn when you use the feature, unless you have explicitly disabled the warning:

```
no warnings "experimental::try";
```

This feature enables the "try" and "catch" syntax, which allows exception handling, where exceptions thrown from the body of the block introduced with "try" are caught by executing the body of the "catch" block.

For more information, see "Try Catch Exception Handling" in perlsyn.

The 'defer' feature

WARNING: This feature is still experimental and the implementation may change or be removed in future versions of Perl. For this reason, Perl will warn when you use the feature, unless you have explicitly disabled the warning:

```
no warnings "experimental::defer";
```

This feature enables the "defer" block syntax, which allows a block of code to be deferred until when the flow of control leaves the block which contained it. For more details, see "defer" in perlsyn.

The 'extra_paired_delimiters' feature

WARNING: This feature is still experimental and the implementation may change or be removed in future versions of Perl. For this reason, Perl will warn when you use the feature, unless you have explicitly disabled the warning:

```
no warnings "experimental::extra_paired_delimiters";
```

This feature enables the use of more paired string delimiters than the traditional four, "< >", "( )", "{ }", and "[ ]". When this feature is on, for example, you can say "qrXpatX".

This feature is available starting in Perl 5.36.

The complete list of accepted paired delimiters as of Unicode 14.0 is:

```
( )   U+0028, U+0029   LEFT/RIGHT PARENTHESIS
< >   U+003C, U+003E   LESS-THAN/GREATER-THAN SIGN
[ ]   U+005B, U+005D   LEFT/RIGHT SQUARE BRACKET
{ }   U+007B, U+007D   LEFT/RIGHT CURLY BRACKET
X X   U+00AB, U+00BB   LEFT/RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK
```

X X U+00BB, U+00AB RIGHT/LEFT-POINTING DOUBLE ANGLE QUOTATION MARK
X X U+0706, U+0707 SYRIAC COLON SKEWED LEFT/RIGHT
X X U+0F3A, U+0F3B TIBETAN MARK GUG RTAGS GYON, TIBETAN MARK GUG RTAGS GYAS
X X U+0F3C, U+0F3D TIBETAN MARK ANG KHANG GYON, TIBETAN MARK ANG KHANG GYAS
X X U+169B, U+169C OGHAM FEATHER MARK, OGHAM REVERSED FEATHER MARK
X X U+2018, U+2019 LEFT/RIGHT SINGLE QUOTATION MARK
X X U+2019, U+2018 RIGHT/LEFT SINGLE QUOTATION MARK
X X U+201C, U+201D LEFT/RIGHT DOUBLE QUOTATION MARK
X X U+201D, U+201C RIGHT/LEFT DOUBLE QUOTATION MARK
X X U+2035, U+2032 REVERSED PRIME, PRIME
X X U+2036, U+2033 REVERSED DOUBLE PRIME, DOUBLE PRIME
X X U+2037, U+2034 REVERSED TRIPLE PRIME, TRIPLE PRIME
X X U+2039, U+203A SINGLE LEFT/RIGHT-POINTING ANGLE QUOTATION MARK
X X U+203A, U+2039 SINGLE RIGHT/LEFT-POINTING ANGLE QUOTATION MARK
X X U+2045, U+2046 LEFT/RIGHT SQUARE BRACKET WITH QUILL
X X U+204D, U+204C BLACK RIGHT/LEFTWARDS BULLET
X X U+207D, U+207E SUPERSCRIPT LEFT/RIGHT PARENTHESIS
X X U+208D, U+208E SUBSCRIPT LEFT/RIGHT PARENTHESIS
X X U+2192, U+2190 RIGHT/LEFTWARDS ARROW
X X U+219B, U+219A RIGHT/LEFTWARDS ARROW WITH STROKE
X X U+219D, U+219C RIGHT/LEFTWARDS WAVE ARROW
X X U+21A0, U+219E RIGHT/LEFTWARDS TWO HEADED ARROW
X X U+21A3, U+21A2 RIGHT/LEFTWARDS ARROW WITH TAIL
X X U+21A6, U+21A4 RIGHT/LEFTWARDS ARROW FROM BAR
X X U+21AA, U+21A9 RIGHT/LEFTWARDS ARROW WITH HOOK
X X U+21AC, U+21AB RIGHT/LEFTWARDS ARROW WITH LOOP
X X U+21B1, U+21B0 UPWARDS ARROW WITH TIP RIGHT/LEFTWARDS
X X U+21B3, U+21B2 DOWNWARDS ARROW WITH TIP RIGHT/LEFTWARDS
X X U+21C0, U+21BC RIGHT/LEFTWARDS HARPOON WITH BARB UPWARDS
X X U+21C1, U+21BD RIGHT/LEFTWARDS HARPOON WITH BARB DOWNWARDS
X X U+21C9, U+21C7 RIGHT/LEFTWARDS PAIRED ARROWS
X X U+21CF, U+21CD RIGHT/LEFTWARDS DOUBLE ARROW WITH STROKE
X X U+21D2, U+21D0 RIGHT/LEFTWARDS DOUBLE ARROW
X X U+21DB, U+21DA RIGHT/LEFTWARDS TRIPLE ARROW
X X U+21DD, U+21DC RIGHT/LEFTWARDS SQUIGGLE ARROW
X X U+21E2, U+21E0 RIGHT/LEFTWARDS DASHED ARROW
X X U+21E5, U+21E4 RIGHT/LEFTWARDS ARROW TO BAR
X X U+21E8, U+21E6 RIGHT/LEFTWARDS WHITE ARROW

X X U+21F4, U+2B30 RIGHT/LEFT ARROW WITH SMALL CIRCLE
X X U+21F6, U+2B31 THREE RIGHT/LEFTWARDS ARROWS
X X U+21F8, U+21F7 RIGHT/LEFTWARDS ARROW WITH VERTICAL STROKE
X X U+21FB, U+21FA RIGHT/LEFTWARDS ARROW WITH DOUBLE VERTICAL STROKE
X X U+21FE, U+21FD RIGHT/LEFTWARDS OPEN-HEADED ARROW
X X U+2208, U+220B ELEMENT OF, CONTAINS AS MEMBER
X X U+2209, U+220C NOT AN ELEMENT OF, DOES NOT CONTAIN AS MEMBER
X X U+220A, U+220D SMALL ELEMENT OF, SMALL CONTAINS AS MEMBER
X X U+2264, U+2265 LESS-THAN/GREATER-THAN OR EQUAL TO
X X U+2266, U+2267 LESS-THAN/GREATER-THAN OVER EQUAL TO
X X U+2268, U+2269 LESS-THAN/GREATER-THAN BUT NOT EQUAL TO
X X U+226A, U+226B MUCH LESS-THAN/GREATER-THAN
X X U+226E, U+226F NOT LESS-THAN/GREATER-THAN
X X U+2270, U+2271 NEITHER LESS-THAN/GREATER-THAN NOR EQUAL TO
X X U+2272, U+2273 LESS-THAN/GREATER-THAN OR EQUIVALENT TO
X X U+2274, U+2275 NEITHER LESS-THAN/GREATER-THAN NOR EQUIVALENT TO
X X U+227A, U+227B PRECEDES/SUCCEEDS
X X U+227C, U+227D PRECEDES/SUCCEEDS OR EQUAL TO
X X U+227E, U+227F PRECEDES/SUCCEEDS OR EQUIVALENT TO
X X U+2280, U+2281 DOES NOT PRECEDE/SUCCEED
X X U+2282, U+2283 SUBSET/SUPERSET OF
X X U+2284, U+2285 NOT A SUBSET/SUPERSET OF
X X U+2286, U+2287 SUBSET/SUPERSET OF OR EQUAL TO
X X U+2288, U+2289 NEITHER A SUBSET/SUPERSET OF NOR EQUAL TO
X X U+228A, U+228B SUBSET/SUPERSET OF WITH NOT EQUAL TO
X X U+22A3, U+22A2 LEFT/RIGHT TACK
X X U+22A6, U+2ADE ASSERTION, SHORT LEFT TACK
X X U+22A8, U+2AE4 TRUE, VERTICAL BAR DOUBLE LEFT TURNSTILE
X X U+22A9, U+2AE3 FORCES, DOUBLE VERTICAL BAR LEFT TURNSTILE
X X U+22B0, U+22B1 PRECEDES/SUCCEEDS UNDER RELATION
X X U+22D0, U+22D1 DOUBLE SUBSET/SUPERSET
X X U+22D6, U+22D7 LESS-THAN/GREATER-THAN WITH DOT
X X U+22D8, U+22D9 VERY MUCH LESS-THAN/GREATER-THAN
X X U+22DC, U+22DD EQUAL TO OR LESS-THAN/GREATER-THAN
X X U+22DE, U+22DF EQUAL TO OR PRECEDES/SUCCEEDS
X X U+22E0, U+22E1 DOES NOT PRECEDE/SUCCEED OR EQUAL
X X U+22E6, U+22E7 LESS-THAN/GREATER-THAN BUT NOT EQUIVALENT TO
X X U+22E8, U+22E9 PRECEDES/SUCCEEDS BUT NOT EQUIVALENT TO
X X U+22F2, U+22FA ELEMENT OF/CONTAINS WITH LONG HORIZONTAL STROKE

X X U+22F3, U+22FB ELEMENT OF/CONTAINS WITH VERTICAL BAR AT END OF HORIZONTAL STROKE
X X U+22F4, U+22FC SMALL ELEMENT OF/CONTAINS WITH VERTICAL BAR AT END OF HORIZONTAL STROKE
X X U+22F6, U+22FD ELEMENT OF/CONTAINS WITH OVERBAR
X X U+22F7, U+22FE SMALL ELEMENT OF/CONTAINS WITH OVERBAR
X X U+2308, U+2309 LEFT/RIGHT CEILING
X X U+230A, U+230B LEFT/RIGHT FLOOR
X X U+2326, U+232B ERASE TO THE RIGHT/LEFT
X X U+2329, U+232A LEFT/RIGHT-POINTING ANGLE BRACKET
X X U+2348, U+2347 APL FUNCTIONAL SYMBOL QUAD RIGHT/LEFTWARDS ARROW
X X U+23E9, U+23EA BLACK RIGHT/LEFT-POINTING DOUBLE TRIANGLE
X X U+23ED, U+23EE BLACK RIGHT/LEFT-POINTING DOUBLE TRIANGLE WITH VERTICAL BAR
X X U+261B, U+261A BLACK RIGHT/LEFT POINTING INDEX
X X U+261E, U+261C WHITE RIGHT/LEFT POINTING INDEX
X X U+269E, U+269F THREE LINES CONVERGING RIGHT/LEFT
X X U+2768, U+2769 MEDIUM LEFT/RIGHT PARENTHESIS ORNAMENT
X X U+276A, U+276B MEDIUM FLATTENED LEFT/RIGHT PARENTHESIS ORNAMENT
X X U+276C, U+276D MEDIUM LEFT/RIGHT-POINTING ANGLE BRACKET ORNAMENT
X X U+276E, U+276F HEAVY LEFT/RIGHT-POINTING ANGLE QUOTATION MARK ORNAMENT
X X U+2770, U+2771 HEAVY LEFT/RIGHT-POINTING ANGLE BRACKET ORNAMENT
X X U+2772, U+2773 LIGHT LEFT/RIGHT TORTOISE SHELL BRACKET ORNAMENT
X X U+2774, U+2775 MEDIUM LEFT/RIGHT CURLY BRACKET ORNAMENT
X X U+27C3, U+27C4 OPEN SUBSET/SUPERSET
X X U+27C5, U+27C6 LEFT/RIGHT S-SHAPED BAG DELIMITER
X X U+27C8, U+27C9 REVERSE SOLIDUS PRECEDING SUBSET, SUPERSET PRECEDING SOLIDUS
X X U+27DE, U+27DD LONG LEFT/RIGHT TACK
X X U+27E6, U+27E7 MATHEMATICAL LEFT/RIGHT WHITE SQUARE BRACKET
X X U+27E8, U+27E9 MATHEMATICAL LEFT/RIGHT ANGLE BRACKET
X X U+27EA, U+27EB MATHEMATICAL LEFT/RIGHT DOUBLE ANGLE BRACKET
X X U+27EC, U+27ED MATHEMATICAL LEFT/RIGHT WHITE TORTOISE SHELL BRACKET
X X U+27EE, U+27EF MATHEMATICAL LEFT/RIGHT FLATTENED PARENTHESIS
X X U+27F4, U+2B32 RIGHT/LEFT ARROW WITH CIRCLED PLUS
X X U+27F6, U+27F5 LONG RIGHT/LEFTWARDS ARROW
X X U+27F9, U+27F8 LONG RIGHT/LEFTWARDS DOUBLE ARROW

X X U+27FC, U+27FB LONG RIGHT/LEFTWARDS ARROW FROM BAR
X X U+27FE, U+27FD LONG RIGHT/LEFTWARDS DOUBLE ARROW FROM BAR
X X U+27FF, U+2B33 LONG RIGHT/LEFTWARDS SQUIGGLE ARROW
X X U+2900, U+2B34 RIGHT/LEFTWARDS TWO-HEADED ARROW WITH VERTICAL STROKE
X X U+2901, U+2B35 RIGHT/LEFTWARDS TWO-HEADED ARROW WITH DOUBLE VERTICAL STROKE
X X U+2903, U+2902 RIGHT/LEFTWARDS DOUBLE ARROW WITH VERTICAL STROKE
X X U+2905, U+2B36 RIGHT/LEFTWARDS TWO-HEADED ARROW FROM BAR
X X U+2907, U+2906 RIGHT/LEFTWARDS DOUBLE ARROW FROM BAR
X X U+290D, U+290C RIGHT/LEFTWARDS DOUBLE DASH ARROW
X X U+290F, U+290E RIGHT/LEFTWARDS TRIPLE DASH ARROW
X X U+2910, U+2B37 RIGHT/LEFTWARDS TWO-HEADED TRIPLE DASH ARROW
X X U+2911, U+2B38 RIGHT/LEFTWARDS ARROW WITH DOTTED STEM
X X U+2914, U+2B39 RIGHT/LEFTWARDS ARROW WITH TAIL WITH VERTICAL STROKE
X X U+2915, U+2B3A RIGHT/LEFTWARDS ARROW WITH TAIL WITH DOUBLE VERTICAL STROKE
X X U+2916, U+2B3B RIGHT/LEFTWARDS TWO-HEADED ARROW WITH TAIL
X X U+2917, U+2B3C RIGHT/LEFTWARDS TWO-HEADED ARROW WITH TAIL WITH VERTICAL STROKE
X X U+2918, U+2B3D RIGHT/LEFTWARDS TWO-HEADED ARROW WITH TAIL WITH DOUBLE VERTICAL STROKE
X X U+291A, U+2919 RIGHT/LEFTWARDS ARROW-TAIL
X X U+291C, U+291B RIGHT/LEFTWARDS DOUBLE ARROW-TAIL
X X U+291E, U+291D RIGHT/LEFTWARDS ARROW TO BLACK DIAMOND
X X U+2920, U+291F RIGHT/LEFTWARDS ARROW FROM BAR TO BLACK DIAMOND
X X U+2933, U+2B3F WAVE ARROW POINTING DIRECTLY RIGHT/LEFT
X X U+2937, U+2936 ARROW POINTING DOWNWARDS THEN CURVING RIGHT/LEFTWARDS
X X U+2945, U+2946 RIGHT/LEFTWARDS ARROW WITH PLUS BELOW
X X U+2947, U+2B3E RIGHT/LEFTWARDS ARROW THROUGH X
X X U+2953, U+2952 RIGHT/LEFTWARDS HARPOON WITH BARB UP TO BAR
X X U+2957, U+2956 RIGHT/LEFTWARDS HARPOON WITH BARB DOWN TO BAR
X X U+295B, U+295A RIGHT/LEFTWARDS HARPOON WITH BARB UP FROM BAR
X X U+295F, U+295E RIGHT/LEFTWARDS HARPOON WITH BARB DOWN FROM BAR
X X U+2964, U+2962 RIGHT/LEFTWARDS HARPOON WITH BARB UP ABOVE RIGHT/LEFTWARDS HARPOON WITH BARB DOWN
X X U+296C, U+296A RIGHT/LEFTWARDS HARPOON WITH BARB UP ABOVE LONG

DASH

X X U+296D, U+296B RIGHT/LEFTWARDS HARPOON WITH BARB DOWN BELOW LONG DASH

X X U+2971, U+2B40 EQUALS SIGN ABOVE RIGHT/LEFTWARDS ARROW

X X U+2972, U+2B41 TILDE OPERATOR ABOVE RIGHTWARDS ARROW, REVERSE TILDE OPERATOR ABOVE LEFTWARDS ARROW

X X U+2974, U+2B4B RIGHTWARDS ARROW ABOVE TILDE OPERATOR, LEFTWARDS ARROW ABOVE REVERSE TILDE OPERATOR

X X U+2975, U+2B42 RIGHTWARDS ARROW ABOVE ALMOST EQUAL TO, LEFTWARDS ARROW ABOVE REVERSE ALMOST EQUAL TO

X X U+2979, U+297B SUBSET/SUPERSET ABOVE RIGHT/LEFTWARDS ARROW

X X U+2983, U+2984 LEFT/RIGHT WHITE CURLY BRACKET

X X U+2985, U+2986 LEFT/RIGHT WHITE PARENTHESIS

X X U+2987, U+2988 Z NOTATION LEFT/RIGHT IMAGE BRACKET

X X U+2989, U+298A Z NOTATION LEFT/RIGHT BINDING BRACKET

X X U+298B, U+298C LEFT/RIGHT SQUARE BRACKET WITH UNDERBAR

X X U+298D, U+2990 LEFT/RIGHT SQUARE BRACKET WITH TICK IN TOP CORNER

X X U+298F, U+298E LEFT/RIGHT SQUARE BRACKET WITH TICK IN BOTTOM CORNER

X X U+2991, U+2992 LEFT/RIGHT ANGLE BRACKET WITH DOT

X X U+2993, U+2994 LEFT/RIGHT ARC LESS-THAN/GREATER-THAN BRACKET

X X U+2995, U+2996 DOUBLE LEFT/RIGHT ARC GREATER-THAN/LESS-THAN BRACKET

X X U+2997, U+2998 LEFT/RIGHT BLACK TORTOISE SHELL BRACKET

X X U+29A8, U+29A9 MEASURED ANGLE WITH OPEN ARM ENDING IN ARROW POINTING UP AND RIGHT/LEFT

X X U+29AA, U+29AB MEASURED ANGLE WITH OPEN ARM ENDING IN ARROW POINTING DOWN AND RIGHT/LEFT

X X U+29B3, U+29B4 EMPTY SET WITH RIGHT/LEFT ARROW ABOVE

X X U+29C0, U+29C1 CIRCLED LESS-THAN/GREATER-THAN

X X U+29D8, U+29D9 LEFT/RIGHT WIGGLY FENCE

X X U+29DA, U+29DB LEFT/RIGHT DOUBLE WIGGLY FENCE

X X U+29FC, U+29FD LEFT/RIGHT-POINTING CURVED ANGLE BRACKET

X X U+2A79, U+2A7A LESS-THAN/GREATER-THAN WITH CIRCLE INSIDE

X X U+2A7B, U+2A7C LESS-THAN/GREATER-THAN WITH QUESTION MARK ABOVE

X X U+2A7D, U+2A7E LESS-THAN/GREATER-THAN OR SLANTED EQUAL TO

X X U+2A7F, U+2A80 LESS-THAN/GREATER-THAN OR SLANTED EQUAL TO WITH DOT INSIDE

X X U+2A81, U+2A82 LESS-THAN/GREATER-THAN OR SLANTED EQUAL TO WITH

DOT ABOVE
X X U+2A83, U+2A84 LESS-THAN/GREATER-THAN OR SLANTED EQUAL TO WITH DOT ABOVE RIGHT/LEFT
X X U+2A85, U+2A86 LESS-THAN/GREATER-THAN OR APPROXIMATE
X X U+2A87, U+2A88 LESS-THAN/GREATER-THAN AND SINGLE-LINE NOT EQUAL TO
X X U+2A89, U+2A8A LESS-THAN/GREATER-THAN AND NOT APPROXIMATE
X X U+2A8D, U+2A8E LESS-THAN/GREATER-THAN ABOVE SIMILAR OR EQUAL
X X U+2A95, U+2A96 SLANTED EQUAL TO OR LESS-THAN/GREATER-THAN
X X U+2A97, U+2A98 SLANTED EQUAL TO OR LESS-THAN/GREATER-THAN WITH DOT INSIDE
X X U+2A99, U+2A9A DOUBLE-LINE EQUAL TO OR LESS-THAN/GREATER-THAN
X X U+2A9B, U+2A9C DOUBLE-LINE SLANTED EQUAL TO OR LESS-THAN/ GREATER-THAN
X X U+2A9D, U+2A9E SIMILAR OR LESS-THAN/GREATER-THAN
X X U+2A9F, U+2AA0 SIMILAR ABOVE LESS-THAN/GREATER-THAN ABOVE EQUALS SIGN
X X U+2AA1, U+2AA2 DOUBLE NESTED LESS-THAN/GREATER-THAN
X X U+2AA6, U+2AA7 LESS-THAN/GREATER-THAN CLOSED BY CURVE
X X U+2AA8, U+2AA9 LESS-THAN/GREATER-THAN CLOSED BY CURVE ABOVE SLANTED EQUAL
X X U+2AAA, U+2AAB SMALLER THAN/LARGER THAN
X X U+2AAC, U+2AAD SMALLER THAN/LARGER THAN OR EQUAL TO
X X U+2AAF, U+2AB0 PRECEDES/SUCCEEDS ABOVE SINGLE-LINE EQUALS SIGN
X X U+2AB1, U+2AB2 PRECEDES/SUCCEEDS ABOVE SINGLE-LINE NOT EQUAL TO
X X U+2AB3, U+2AB4 PRECEDES/SUCCEEDS ABOVE EQUALS SIGN
X X U+2AB5, U+2AB6 PRECEDES/SUCCEEDS ABOVE NOT EQUAL TO
X X U+2AB7, U+2AB8 PRECEDES/SUCCEEDS ABOVE ALMOST EQUAL TO
X X U+2AB9, U+2ABA PRECEDES/SUCCEEDS ABOVE NOT ALMOST EQUAL TO
X X U+2ABB, U+2ABC DOUBLE PRECEDES/SUCCEEDS
X X U+2ABD, U+2ABE SUBSET/SUPERSET WITH DOT
X X U+2ABF, U+2AC0 SUBSET/SUPERSET WITH PLUS SIGN BELOW
X X U+2AC1, U+2AC2 SUBSET/SUPERSET WITH MULTIPLICATION SIGN BELOW
X X U+2AC3, U+2AC4 SUBSET/SUPERSET OF OR EQUAL TO WITH DOT ABOVE
X X U+2AC5, U+2AC6 SUBSET/SUPERSET OF ABOVE EQUALS SIGN
X X U+2AC7, U+2AC8 SUBSET/SUPERSET OF ABOVE TILDE OPERATOR
X X U+2AC9, U+2ACA SUBSET/SUPERSET OF ABOVE ALMOST EQUAL TO
X X U+2ACB, U+2ACC SUBSET/SUPERSET OF ABOVE NOT EQUAL TO
X X U+2ACF, U+2AD0 CLOSED SUBSET/SUPERSET
X X U+2AD1, U+2AD2 CLOSED SUBSET/SUPERSET OR EQUAL TO

X X U+2AD5, U+2AD6 SUBSET/SUPERSET ABOVE SUBSET/SUPERSET
X X U+2AE5, U+22AB DOUBLE VERTICAL BAR DOUBLE LEFT/RIGHT TURNSTILE
X X U+2AF7, U+2AF8 TRIPLE NESTED LESS-THAN/GREATER-THAN
X X U+2AF9, U+2AFA DOUBLE-LINE SLANTED LESS-THAN/GREATER-THAN OR EQUAL TO
X X U+2B46, U+2B45 RIGHT/LEFTWARDS QUADRUPLE ARROW
X X U+2B47, U+2B49 REVERSE TILDE OPERATOR ABOVE RIGHTWARDS ARROW, TILDE OPERATOR ABOVE LEFTWARDS ARROW
X X U+2B48, U+2B4A RIGHTWARDS ARROW ABOVE REVERSE ALMOST EQUAL TO, LEFTWARDS ARROW ABOVE ALMOST EQUAL TO
X X U+2B4C, U+2973 RIGHTWARDS ARROW ABOVE REVERSE TILDE OPERATOR, LEFTWARDS ARROW ABOVE TILDE OPERATOR
X X U+2B62, U+2B60 RIGHT/LEFTWARDS TRIANGLE-HEADED ARROW
X X U+2B6C, U+2B6A RIGHT/LEFTWARDS TRIANGLE-HEADED DASHED ARROW
X X U+2B72, U+2B70 RIGHT/LEFTWARDS TRIANGLE-HEADED ARROW TO BAR
X X U+2B7C, U+2B7A RIGHT/LEFTWARDS TRIANGLE-HEADED ARROW WITH DOUBLE VERTICAL STROKE
X X U+2B86, U+2B84 RIGHT/LEFTWARDS TRIANGLE-HEADED PAIRED ARROWS
X X U+2B8A, U+2B88 RIGHT/LEFTWARDS BLACK CIRCLED WHITE ARROW
X X U+2B95, U+2B05 RIGHT/LEFTWARDS BLACK ARROW
X X U+2B9A, U+2B98 THREE-D TOP-LIGHTED RIGHT/LEFTWARDS EQUILATERAL ARROWHEAD
X X U+2B9E, U+2B9C BLACK RIGHT/LEFTWARDS EQUILATERAL ARROWHEAD
X X U+2BA1, U+2BA0 DOWNWARDS TRIANGLE-HEADED ARROW WITH LONG TIP RIGHT/LEFTWARDS
X X U+2BA3, U+2BA2 UPWARDS TRIANGLE-HEADED ARROW WITH LONG TIP RIGHT/LEFTWARDS
X X U+2BA9, U+2BA8 BLACK CURVED DOWNWARDS AND RIGHT/LEFTWARDS ARROW
X X U+2BAB, U+2BAA BLACK CURVED UPWARDS AND RIGHT/LEFTWARDS ARROW
X X U+2BB1, U+2BB0 RIBBON ARROW DOWN RIGHT/LEFT
X X U+2BB3, U+2BB2 RIBBON ARROW UP RIGHT/LEFT
X X U+2BEE, U+2BEC RIGHT/LEFTWARDS TWO-HEADED ARROW WITH TRIANGLE ARROWHEADS
X X U+2E02, U+2E03 LEFT/RIGHT SUBSTITUTION BRACKET
X X U+2E03, U+2E02 RIGHT/LEFT SUBSTITUTION BRACKET
X X U+2E04, U+2E05 LEFT/RIGHT DOTTED SUBSTITUTION BRACKET
X X U+2E05, U+2E04 RIGHT/LEFT DOTTED SUBSTITUTION BRACKET
X X U+2E09, U+2E0A LEFT/RIGHT TRANSPOSITION BRACKET
X X U+2E0A, U+2E09 RIGHT/LEFT TRANSPOSITION BRACKET
X X U+2E0C, U+2E0D LEFT/RIGHT RAISED OMISSION BRACKET

X X U+2E0D, U+2E0C RIGHT/LEFT RAISED OMISSION BRACKET
X X U+2E11, U+2E10 REVERSED FORKED PARAGRAPHOS, FORKED PARAGRAPHOS
X X U+2E1C, U+2E1D LEFT/RIGHT LOW PARAPHRASE BRACKET
X X U+2E1D, U+2E1C RIGHT/LEFT LOW PARAPHRASE BRACKET
X X U+2E20, U+2E21 LEFT/RIGHT VERTICAL BAR WITH QUILL
X X U+2E21, U+2E20 RIGHT/LEFT VERTICAL BAR WITH QUILL
X X U+2E22, U+2E23 TOP LEFT/RIGHT HALF BRACKET
X X U+2E24, U+2E25 BOTTOM LEFT/RIGHT HALF BRACKET
X X U+2E26, U+2E27 LEFT/RIGHT SIDEWAYS U BRACKET
X X U+2E28, U+2E29 LEFT/RIGHT DOUBLE PARENTHESIS
X X U+2E36, U+2E37 DAGGER WITH LEFT/RIGHT GUARD
X X U+2E42, U+201E DOUBLE LOW-REVERSED-9 QUOTATION MARK, DOUBLE LOW-9 QUOTATION MARK
X X U+2E55, U+2E56 LEFT/RIGHT SQUARE BRACKET WITH STROKE
X X U+2E57, U+2E58 LEFT/RIGHT SQUARE BRACKET WITH DOUBLE STROKE
X X U+2E59, U+2E5A TOP HALF LEFT/RIGHT PARENTHESIS
X X U+2E5B, U+2E5C BOTTOM HALF LEFT/RIGHT PARENTHESIS
X X U+3008, U+3009 LEFT/RIGHT ANGLE BRACKET
X X U+300A, U+300B LEFT/RIGHT DOUBLE ANGLE BRACKET
X X U+300C, U+300D LEFT/RIGHT CORNER BRACKET
X X U+300E, U+300F LEFT/RIGHT WHITE CORNER BRACKET
X X U+3010, U+3011 LEFT/RIGHT BLACK LENTICULAR BRACKET
X X U+3014, U+3015 LEFT/RIGHT TORTOISE SHELL BRACKET
X X U+3016, U+3017 LEFT/RIGHT WHITE LENTICULAR BRACKET
X X U+3018, U+3019 LEFT/RIGHT WHITE TORTOISE SHELL BRACKET
X X U+301A, U+301B LEFT/RIGHT WHITE SQUARE BRACKET
X X U+301D, U+301E REVERSED DOUBLE PRIME QUOTATION MARK, DOUBLE PRIME QUOTATION MARK
X X U+A9C1, U+A9C2 JAVANESE LEFT/RIGHT RERENGGAN
X X U+FD3E, U+FD3F ORNATE LEFT/RIGHT PARENTHESIS
X X U+FE59, U+FE5A SMALL LEFT/RIGHT PARENTHESIS
X X U+FE5B, U+FE5C SMALL LEFT/RIGHT CURLY BRACKET
X X U+FE5D, U+FE5E SMALL LEFT/RIGHT TORTOISE SHELL BRACKET
X X U+FE64, U+FE65 SMALL LESS-THAN/GREATER-THAN SIGN
X X U+FF08, U+FF09 FULLWIDTH LEFT/RIGHT PARENTHESIS
X X U+FF1C, U+FF1E FULLWIDTH LESS-THAN/GREATER-THAN SIGN
X X U+FF3B, U+FF3D FULLWIDTH LEFT/RIGHT SQUARE BRACKET
X X U+FF5B, U+FF5D FULLWIDTH LEFT/RIGHT CURLY BRACKET
X X U+FF5F, U+FF60 FULLWIDTH LEFT/RIGHT WHITE PARENTHESIS
X X U+FF62, U+FF63 HALFWIDTH LEFT/RIGHT CORNER BRACKET

X X U+FFEB, U+FFE9 HALFWIDTH RIGHT/LEFTWARDS ARROW
X X U+1D103, U+1D102 MUSICAL SYMBOL REVERSE FINAL BARLINE, MUSICAL SYMBOL FINAL BARLINE
X X U+1D106, U+1D107 MUSICAL SYMBOL LEFT/RIGHT REPEAT SIGN
X X U+1F449, U+1F448 WHITE RIGHT/LEFT POINTING BACKHAND INDEX
X X U+1F508, U+1F568 SPEAKER, RIGHT SPEAKER
X X U+1F509, U+1F569 SPEAKER WITH ONE SOUND WAVE, RIGHT SPEAKER WITH ONE SOUND WAVE
X X U+1F50A, U+1F56A SPEAKER WITH THREE SOUND WAVES, RIGHT SPEAKER WITH THREE SOUND WAVES
X X U+1F57B, U+1F57D LEFT/RIGHT HAND TELEPHONE RECEIVER
X X U+1F599, U+1F598 SIDEWAYS WHITE RIGHT/LEFT POINTING INDEX
X X U+1F59B, U+1F59A SIDEWAYS BLACK RIGHT/LEFT POINTING INDEX
X X U+1F59D, U+1F59C BLACK RIGHT/LEFT POINTING BACKHAND INDEX
X X U+1F5E6, U+1F5E7 THREE RAYS LEFT/RIGHT
X X U+1F802, U+1F800 RIGHT/LEFTWARDS ARROW WITH SMALL TRIANGLE ARROWHEAD
X X U+1F806, U+1F804 RIGHT/LEFTWARDS ARROW WITH MEDIUM TRIANGLE ARROWHEAD
X X U+1F80A, U+1F808 RIGHT/LEFTWARDS ARROW WITH LARGE TRIANGLE ARROWHEAD
X X U+1F812, U+1F810 RIGHT/LEFTWARDS ARROW WITH SMALL EQUILATERAL ARROWHEAD
X X U+1F816, U+1F814 RIGHT/LEFTWARDS ARROW WITH EQUILATERAL ARROWHEAD
X X U+1F81A, U+1F818 HEAVY RIGHT/LEFTWARDS ARROW WITH EQUILATERAL ARROWHEAD
X X U+1F81E, U+1F81C HEAVY RIGHT/LEFTWARDS ARROW WITH LARGE EQUILATERAL ARROWHEAD
X X U+1F822, U+1F820 RIGHT/LEFTWARDS TRIANGLE-HEADED ARROW WITH NARROW SHAFT
X X U+1F826, U+1F824 RIGHT/LEFTWARDS TRIANGLE-HEADED ARROW WITH MEDIUM SHAFT
X X U+1F82A, U+1F828 RIGHT/LEFTWARDS TRIANGLE-HEADED ARROW WITH BOLD SHAFT
X X U+1F82E, U+1F82C RIGHT/LEFTWARDS TRIANGLE-HEADED ARROW WITH HEAVY SHAFT
X X U+1F832, U+1F830 RIGHT/LEFTWARDS TRIANGLE-HEADED ARROW WITH VERY HEAVY SHAFT
X X U+1F836, U+1F834 RIGHT/LEFTWARDS FINGER-POST ARROW
X X U+1F83A, U+1F838 RIGHT/LEFTWARDS SQUARED ARROW

X X U+1F83E, U+1F83C RIGHT/LEFTWARDS COMPRESSED ARROW
X X U+1F842, U+1F840 RIGHT/LEFTWARDS HEAVY COMPRESSED ARROW
X X U+1F846, U+1F844 RIGHT/LEFTWARDS HEAVY ARROW
X X U+1F852, U+1F850 RIGHT/LEFTWARDS SANS-SERIF ARROW
X X U+1F862, U+1F860 WIDE-HEADED RIGHT/LEFTWARDS LIGHT BARB ARROW
X X U+1F86A, U+1F868 WIDE-HEADED RIGHT/LEFTWARDS BARB ARROW
X X U+1F872, U+1F870 WIDE-HEADED RIGHT/LEFTWARDS MEDIUM BARB ARROW
X X U+1F87A, U+1F878 WIDE-HEADED RIGHT/LEFTWARDS HEAVY BARB ARROW
X X U+1F882, U+1F880 WIDE-HEADED RIGHT/LEFTWARDS VERY HEAVY BARB ARROW
X X U+1F892, U+1F890 RIGHT/LEFTWARDS TRIANGLE ARROWHEAD
X X U+1F896, U+1F894 RIGHT/LEFTWARDS WHITE ARROW WITHIN TRIANGLE ARROWHEAD
X X U+1F89A, U+1F898 RIGHT/LEFTWARDS ARROW WITH NOTCHED TAIL
X X U+1F8A1, U+1F8A0 RIGHTWARDS BOTTOM SHADED WHITE ARROW, LEFTWARDS BOTTOM-SHADED WHITE ARROW
X X U+1F8A3, U+1F8A2 RIGHT/LEFTWARDS TOP SHADED WHITE ARROW
X X U+1F8A5, U+1F8A6 RIGHT/LEFTWARDS RIGHT-SHADED WHITE ARROW
X X U+1F8A7, U+1F8A4 RIGHT/LEFTWARDS LEFT-SHADED WHITE ARROW
X X U+1F8A9, U+1F8A8 RIGHT/LEFTWARDS BACK-TILTED SHADOWED WHITE ARROW
X X U+1F8AB, U+1F8AA RIGHT/LEFTWARDS FRONT-TILTED SHADOWED WHITE ARROW

FEATURE BUNDLES

It's possible to load multiple features together, using a *feature bundle*. The name of a feature bundle is prefixed with a colon, to distinguish it from an actual feature.

```
use feature ":5.10";
```

The following feature bundles are available:

```
bundle    features included
--------- -----------------
:default  indirect multidimensional
          bareword_filehandles

:5.10     bareword_filehandles indirect
          multidimensional say state switch

:5.12     bareword_filehandles indirect
```

multidimensional say state switch
unicode_strings

:5.14 bareword_filehandles indirect
multidimensional say state switch
unicode_strings

:5.16 bareword_filehandles current_sub evalbytes
fc indirect multidimensional say state
switch unicode_eval unicode_strings

:5.18 bareword_filehandles current_sub evalbytes
fc indirect multidimensional say state
switch unicode_eval unicode_strings

:5.20 bareword_filehandles current_sub evalbytes
fc indirect multidimensional say state
switch unicode_eval unicode_strings

:5.22 bareword_filehandles current_sub evalbytes
fc indirect multidimensional say state
switch unicode_eval unicode_strings

:5.24 bareword_filehandles current_sub evalbytes
fc indirect multidimensional postderef_qq
say state switch unicode_eval
unicode_strings

:5.26 bareword_filehandles current_sub evalbytes
fc indirect multidimensional postderef_qq
say state switch unicode_eval
unicode_strings

:5.28 bareword_filehandles bitwise current_sub
evalbytes fc indirect multidimensional
postderef_qq say state switch unicode_eval
unicode_strings

:5.30 bareword_filehandles bitwise current_sub
evalbytes fc indirect multidimensional

postderef_qq say state switch unicode_eval
unicode_strings

:5.32 bareword_filehandles bitwise current_sub
evalbytes fc indirect multidimensional
postderef_qq say state switch unicode_eval
unicode_strings

:5.34 bareword_filehandles bitwise current_sub
evalbytes fc indirect multidimensional
postderef_qq say state switch unicode_eval
unicode_strings

:5.36 bareword_filehandles bitwise current_sub
evalbytes fc isa postderef_qq say signatures
state unicode_eval unicode_strings

The ":default" bundle represents the feature set that is enabled before any "use feature" or "no feature" declaration.

Specifying sub-versions such as the 0 in 5.14.0 in feature bundles has no effect. Feature bundles are guaranteed to be the same for all sub-versions.

```
use feature ":5.14.0";    # same as ":5.14"
use feature ":5.14.1";    # same as ":5.14"
```

IMPLICIT LOADING

Instead of loading feature bundles by name, it is easier to let Perl do implicit loading of a feature bundle for you.

There are two ways to load the "feature" pragma implicitly:

- By using the "-E" switch on the Perl command-line instead of "-e". That will enable the feature bundle for that version of Perl in the main compilation unit (that is, the one-liner that follows "-E").

- By explicitly requiring a minimum Perl version number for your program, with the "use VERSION" construct. That is,

  use v5.10.0;

will do an implicit

```
no feature ':all';
use feature ':5.10';
```

and so on. Note how the trailing sub-version is automatically stripped from the version.

But to avoid portability warnings (see "use" in perlfunc), you may prefer:

```
use 5.010;
```

with the same effect.

If the required version is older than Perl 5.10, the ":default" feature bundle is automatically loaded instead.

Unlike "use feature ":5.12"", saying "use v5.12" (or any higher version) also does the equivalent of "use strict"; see "use" in perlfunc for details.

CHECKING FEATURES

"feature" provides some simple APIs to check which features are enabled.

These functions cannot be imported and must be called by their fully qualified names. If you don't otherwise need to set a feature you will need to ensure "feature" is loaded with:

```
use feature ();
```

feature_enabled($feature)
feature_enabled($feature, $depth)

```
package MyStandardEnforcer;
use feature ();
use Carp "croak";
sub import {
 croak "disable indirect!" if feature::feature_enabled("indirect");
}
```

Test whether a named feature is enabled at a given level in the call stack, returning a true value if it is. $depth defaults to 1, which checks the scope that called the scope calling **feature::feature_enabled()**.

croaks for an unknown feature name.

features_enabled()

features_enabled($depth)

```
package ReportEnabledFeatures;
use feature "say";
sub import {
  say STDERR join " ", feature::features_enabled();
}
```

Returns a list of the features enabled at a given level in the call stack. $depth defaults to 1, which checks the scope that called the scope calling **feature::features_enabled()**.

feature_bundle()

feature_bundle($depth)

Returns the feature bundle, if any, selected at a given level in the call stack. $depth defaults to 1, which checks the scope that called the scope calling **feature::feature_bundle()**.

Returns an undefined value if no feature bundle is selected in the scope.

The bundle name returned will be for the earliest bundle matching the selected bundle, so:

```
use feature ();
use v5.12;
BEGIN { print feature::feature_bundle(0); }
```

will print 5.11.

This returns internal state, at this point "use v5.12;" sets the feature bundle, but " use feature ":5.12"; " does not set the feature bundle. This may change in a future release of perl.