

perldiag

Table des matières

1	NAME/NOM	1
2	DESCRIPTION	1
3	TRADUCTION	33
3.1	Version	33
3.2	Traducteurs	33
3.3	Relecture	33
4	À propos de ce document	33

1 NAME/NOM

perldiag - Les différents messages de Perl

2 DESCRIPTION

Les différents messages sont classés comme suit (listés dans l'ordre croissant d'importance) :

- (W) Avertissement (optionnel).
- (D) Désapprobation (optionnel).
- (S) Avertissement sérieux (impératif).
- (F) Erreur fatale capturable.
- (P) Erreur interne que vous ne verrez probablement jamais (capturable).
- (X) Erreur fatale non capturable
- (A) Erreur externe (non générée par Perl).

Les messages optionnels sont activés par l'option **-w**. Les avertissements peuvent être capturés en faisant pointer `$_SIG{__WARN__}` vers une référence sur une fonction qui sera appelée à chaque avertissement au lieu d'être fichier. Voir *perlvar*. Les erreurs qui peuvent être capturées par l'opérateur `eval`. Voir `eval()` dans *perlfunc*.

Certains des messages sont génériques. Les endroits qui changent sont notés `%s`, comme dans le format de `printf`. Remarquez que certains messages commencent par `%s!` Les symboles `%(-?@` trient avant les lettres, alors que `[` et `\` trient après.

"my" variable %s can't be in a package

(F) La portée des variables n'est pas dans le paquetage, et cela n'a pas de sens d'essayer d'en déclarer une avec le nom du paquetage devant. Utiliser `local()` si vous voulez localiser une variable de paquetage.

"my" variable %s masks earlier declaration in same scope

(W) Une variable a été redéclarée dans le même bloc, ce qui élimine tous les accès à l'instance précédente. C'est le plus souvent une faute de frappe. Remarquez que la variable déclarée plus tôt continue d'exister en silence jusqu'à la fin du bloc ou jusqu'à ce que les références vers elle soient détruites.

"no" not allowed in expression

(F) Le mot clé « no » a été trouvé et exécuté à la compilation, et retourne une valeur non utilisable. Voir *perlmod*.

"use" not allowed in expression

(F) Le mot clé « use » a été trouvé et exécuté à la compilation et retourne une valeur non utilisable. Voir *perlmod*.

% may only be used in unpack

(F) Vous ne pouvez compacter une chaîne en soumettant un checksum, car le processus de checksum perd l'information, et vous ne pouvez aller dans l'autre sens. Voir `unpack()` dans *perlfunc*.

%s (...) interpreted as function

(W) Vous tombez dans la loi qui dit que chaque liste d'opérateurs suivie par des parenthèses est transformée en fonction, avec toute la liste d'arguments trouvés dans les parenthèses. Voir Termes et opérateurs de listes (leftward) in *perlop*.

%s argument is not a HASH element

(F) L'argument de `exists()` doit être un élément d'un tableau de hachage, comme

```
$foo{$bar}
$ref->[12]->{"susie"}
```

%s argument is not a HASH element or slice

(F) L'argument de `delete()` doit être ou un élément d'un tableau de hachage, comme

```
$foo{$bar}
$ref->[12]->{"susie"}
```

ou une partie d'un tableau de référence comme

```
@foo{$bar, $baz, $xyzyz}
@{$ref->[12]}{"susie", "queue"}
```

%s did not return a true value

(F) Une librairie (ou un fichier utilisé) doit retourner une valeur vraie pour indiquer que la compilation est correcte et que l'initialisation de son code s'est bien passé. Il est habituel de placer « 1; » en fin de fichier, ou une quelconque valeur vraie. Voir `require()` dans *perlfunc*.

%s found where operator expected

(S) L'analyseur syntaxique de Perl sait si il attend un terme ou un opérateur. Si il voit ce qu'il sait être un terme alors qu'il s'attend à un opérateur, il donne ce message d'alerte. Habituellement, cela indique qu'un opérateur a été omis, comme une point virgule.

%s had compilation errors

(F) C'est le message final lorsque `perl -c` échoue.

%s has too many errors

(F) L'analyseur rend la main après 10 erreurs. Les messages suivants ne seraient pas sensés.

%s matches null string many times

(W) L'exemple que vous donnez entre dans une boucle infinie si le moteur d'expression rationnelle ne contrôle pas cela. Voir *perlre*.

%s never introduced

(S) Le symbole en question a été déclaré mais hors de portée avant qu'il soit possible de s'en servir.

%s syntax OK

(F) Message final lorsque `perl -c` réussit.

%s: Command not found

(A) Vous lancez accidentellement votre script par **csh** au lieu de Perl. Vérifiez la ligne avec `#!`, ou lancez votre script manuellement dans Perl.

%s: Expression syntax

(A) Vous lancez accidentellement votre script par **csh** au lieu de Perl. Vérifiez la ligne avec `#!`, ou lancez votre script manuellement dans Perl.

%s: Undefined variable

(A) Vous lancez accidentellement votre script par **csh** au lieu de Perl. Vérifiez la ligne avec `#!`, ou lancez votre script manuellement dans Perl.

%s: not found

(A) Vous lancez accidentellement votre script par le Bourne shell au lieu de Perl. Vérifiez la ligne avec `#!`, ou lancez votre script manuellement dans Perl.

(Missing semicolon on previous line?)

(S) C'est un message donne a titre indicatif en complément avec le message « %s found where operator expected ». Oublier un point virgule déclenche automatiquement ce message.

-P not allowed for setuid/setgid script

(F) Le script a voulu être ouvert par le préprocesseur C par son nom, ce qui entraîne une erreur d'exécution qui casse la sécurité.

-T and -B not implemented on filehandles

(F) Perl ne peut utiliser le tampon de sortie du descripteur de fichier quand il ne connaît pas le type de sortie. Vous devez utiliser un nom de fichier à la place.

-p destination: %s

(F) Une erreur est survenue sur la sortie standard invoquée par l'option -p. (Cette sortie est redirigée vers STDOUT à moins que vous la redirigiez avec select().)

?+* follows nothing in regex

(F) Vous commencez une expression rationnelle avec un quantifiant. Mettre un '\ ' si vous pensez au sens littéraire. Voir *perlre*.

@ outside of string

(F) Vous avez une enveloppe temporaire qui spécifie une position absolue en dehors de la chaîne à décompacter. Voir `pack()` dans *perlfunc*.

accept() on closed fd

(W) Vous tentez de faire un `accept()` sur une socket fermée. Peut-être avez-vous oublié de vérifier la valeur retournée par l'appel de `socket()`? Voir `accept()` dans *perlfunc*.

Allocation too large: %lx

(X) Vous ne pouvez allouer plus de 64K sur machine MS-DOS.

Applying %s to %s will act on scalar(%s)

(W) Les expressions de comparaison (*//*), substitution (*s///*), et translation (*tr///*) fonctionnent avec des valeurs scalaires. Si vous appliquez l'une d'elles sur un tableau ou un tableau de hachage, cela convertit le tableau en une valeur scalaire – la longueur du tableau ou les informations de peuplement du tableau de hachage – puis travaille sur la valeur scalaire. Ce n'est probablement pas ce que vous pensez faire. Voir `grep()` et `map()` dans *perlfunc* pour les alternatives.

Arg too short for msgsnd

(F) `msgsnd()` nécessite une chaîne au moins aussi longue que `sizeof(long)`.

Ambiguous use of %s resolved as %s

(W)(S) Vous exprimez quelque chose qui n'est pas interprété comme tel. Normalement, il est facile de clarifier la situation en ajoutant une quote manquante, un opérateur, une paire de parenthèses ou une déclaration.

Ambiguous call resolved as CORE::%(s), qualify as such or use &

(W) Une fonction que vous avez déclarée a le même nom qu'un mot-clef de Perl, et vous avez utilisé ce nom sans que l'on puisse faire de distinction entre l'un ou l'autre. Perl décide d'appeler la fonction interne car votre fonction n'est pas importée.

Pour forcer l'interprétation sur l'appel de votre fonction, soit vous mettez un ampersand avant le nom, soit vous appelez votre fonction précédée par son nom de paquetage. Alternativement, vous pouvez importer vos fonctions (ou prétendre le faire avec `use subs`).

Pour l'interpréter en silence comme un opérateur Perl, utilisez le préfixe `CORE::` sur l'opérateur (ex. `CORE::log($x)`) ou déclarer la fonction comme une méthode objet (voir *attrs*).

Args must match #! line

(F) L'émulateur `setuid` nécessite que les arguments évoqués correspondent avec ceux utilisés sur la ligne `#!`. Comme certains systèmes imposent un unique argument sur la ligne `#!`, essayez de combiner les opérateurs ; exemple, passez `-w -U` en `-wU`.

Argument "%s" isn't numeric %s

(W) La chaîne indiquée est utilisée comme un argument avec un opérateur qui s'attend à une valeur numérique. Si vous êtes chanceux, le message indique quel opérateur pose problème.

Array @%s missing the @ in argument %d of %s()

(D) Les versions vraiment antérieures de Perl permettaient d'omettre le @ des tableaux à certains endroits. Ceci est maintenant largement obsolète.

assertion botched: %s

(P) Le paquetage `malloc` fourni avec Perl a subi une erreur interne.

Assertion failed: file "%s"

(P) Une affirmation générale a échoué. Le fichier en question doit être examiné.

Assignment to both a list and a scalar

(F) Si vous affectez à un opérateur conditionnel, les 2^e et 3^e arguments doivent ou être tous les deux des scalaires ou tous les deux des listes. Autrement, Perl ne connaît pas le contexte pour fournir le bon coté.

Attempt to free non-arena SV: 0x%lx

(P) Tous les objets SV sont supposés être alloués dans un espace qui sera nettoyé à la sortie du script. Un SV a été découvert en dehors de cet espace.

Attempt to free nonexistent shared string

(P) Perl maintient une table interne qui compte les références de chaînes pour optimiser le stockage et l'accès au tableau associatif et autres chaînes. Cela indique que quelqu'un essaye de décrémenter le compte de références d'un chaîne qui ne serait plus trouvée dans la table.

Attempt to free temp prematurely

(W) Les valeur « Mortalized » sont supposées être libérées par la fonction `free_tmpls()`. Cela indique que quelque chose d'autre a libéré le SV avant que la fonction `free_tmpls()` ait une chance de le faire, ce qui veut dire que la fonction `free_tmpls()` a libéré un scalaire non référencé au moment où vous essayez de le faire.

Attempt to free unreferenced glob pointers

(P) The reference counts got screwed up on symbol aliases.(???)

Attempt to free unreferenced scalar

(W) Perl va décrémenter une référence d'un compteur d'un scalaire pour voir si il arrive à 0, et découvre qu'il est déjà arrivé à 0 plus tôt, et qu'il se peut qu'il ait été libéré par `free()`, et en fait, il a probablement été libéré. Cela peut indiquer que `SvREFCNT_dec()` a été appelé trop de fois, ou que `SvREFCNT_inc()` a été appelé un trop petit nombre de fois, ou que le SV a été mortalized quand il n'aurait pas dû, ou que la mémoire a été corrompue.

Attempt to pack pointer to temporary value

(W) Vous essayez de passer une valeur temporelle (comme le résultat d'une fonction, ou d'une expression calculée) au « p » temporaire de `pack()`. Cela veut dire que le résultat contient un pointeur vers un endroit qui peut devenir invalide à tout moment, même avant la fin de l'expression en cours. Utilisez des valeurs littérales ou globales comme arguments du « p » temporaire de `pack` pour éviter ce message.

Attempt to use reference as lvalue in substr

(W) Vous soumettez une référence comme premier argument de `substr()` utilisé comme un lvalue, ce qui est vraiment étrange. Peut-être avez-vous oublié de la déréréferencer en premier. Voir `substr()` dans *perlfunc*.

Bad arg length for %s, is %d, should be %d

(F) Vous passez un tampon de taille incorrecte à `msgctl()`, `semctl()` ou `shmctl()`. En C, la taille correcte est respectivement `sizeof(struct msqid_ds *)`, `sizeof(struct semid_ds *)`, et `sizeof(struct shmid_ds *)`.

Bad filehandle: %s

(F) Un symbole a été passé à quelque chose qui s'attend à un descripteur de fichier, mais le symbole n'a aucun descripteur associé. Peut-être n'avez-vous pas fait de `open()`, ou dans un autre paquetage.

Bad free() ignored

(S) Une routine interne a appelé `free()` sur quelque chose qui n'a jamais été `malloc()`é dans un premier temps. Obligatoire, mais peut être désactivé en positionnant la variable d'environnement `PERL_BADFREE` à 1.

Ce message peut être assez fréquent avec un fichier `DB_file` sur les systèmes avec les bibliothèques dynamiques en « dur », comme AIX et OS/2. C'est un bug de Berkeley DB qui n'est pas spécifié si vous utilisez la fonction système *forgiving* `malloc()`.

Bad hash

(P) Une des fonctions internes de hash a passé un pointeur HV null.

Bad index while coercing array into hash

(F) L'index a regardé dans le tableau associatif et a découvert que le 0^è élément du pseudo-tableau est illégal. Les valeurs d'index doivent être égales ou supérieures à 1. Voir *perlref*.

Bad name after %s::

(F) Vous démarrez un nom de symbole en utilisant un préfixe de paquetage, et vous n'avez pas fini le symbole. En particulier, vous ne pouvez pas interpoler les deux-points.

Donc

```
$var = 'myvar';
$sym = mypack::$var;
```

n'est pas la même chose que

```
$var = 'myvar';
$sym = "mypack::$var";
```

Bad symbol for array

(P) Une requête interne a voulu ajouter une entrée de tableau à quelque chose qui n'est pas un symbole d'entrée de tableau.

Bad symbol for filehandle

(P) Une requête interne a voulu ajouter un descripteur de fichier à quelque chose qui n'est pas un symbole d'entrée de tableau.

Bad symbol for hash

(P) Une requête interne a voulu ajouter un élément d'un tableau de hachage à quelque chose qui n'est pas un symbole d'entrée de tableau.

Badly placed ()'s

(A) Vous lancez accidentellement votre script par `csh` au lieu de Perl. Vérifiez la ligne avec `#!`, ou lancez votre script manuellement dans Perl.

Bareword "%s" not allowed while "strict subs" in use

(F) Avec "strict subs" en utilisation, un mot est seulement autorisé comme identifiant de fonction, dans la boucle en cours, ou à la gauche du symbole « => ». Peut-être devriez-vous déclarer au préalable votre sous-programme ?

Bareword "%s" refers to nonexistent package

(W) Vous utilisez un mot qualifiant de la forme `FOO::`, mais le compilateur dit qu'il n'y pas d'autres utilisations de ce nom avant ce point. Peut-être devriez-vous déclarer au préalable votre paquetage ?

BEGIN failed—compilation aborted

(F) Une exception non capturable a été levée pendant l'exécution d'un sous-programme `BEGIN`. La compilation stoppe immédiatement et l'interpréteur s'arrête.

BEGIN not safe after errors—compilation aborted

(F) Perl a trouvé un sous-programme `BEGIN {}` (ou une directive `use`, qui implique un `BEGIN {}`) après qu'une ou plusieurs erreurs soient déjà survenues. Tant que l'environnement du `BEGIN {}` ne peut-être garanti (dû aux erreurs), et tant que le code qui suit dépend d'une opération correcte, Perl rend la main.

bind() on closed fd

(W) Vous essayez de faire un `bind()` sur une socket fermée. Peut-être avez-vous oublié de vérifier la valeur retournée par l'appel de `socket()` ? Voir `bind()` dans *perlfunc*.

Bizarre copy of %s in %s

(P) Perl a détecté une tentative de copie d'une valeur interne qui n'est pas copiable.

Callback called exit

(F) Une fonction invoquée depuis un paquetage externe via `perl_call_sv()` s'est terminée en appelant `exit`.

Can't "goto" outside a block

(F) Un `goto` a été exécuté pour sauter ce qui semble être un bloc, excepté que ce n'est pas le bon bloc. Cela apparaît habituellement si vous essayez de sauter hors d'un bloc `sort()` ou d'une fonction, ce qui ne pas de sens. Voir à `goto()` dans *perlfunc*.

Can't "goto" into the middle of a foreach loop

(F) Un « `goto` » a été exécuté pour sauter au milieu d'une boucle `foreach`. Vous ne pouvez y aller depuis l'endroit où vous êtes. Voir `goto()` dans *perlfunc*.

Can't "last" outside a block

(F) L'état « `last` » a été exécuté pour sortir du bloc courant, excepté qu'il n'y a pas de bloc courant. Remarquez que les blocs « `if` » ou « `else` » ne comptent pas comme des blocs « `loop` », ou comme un bloc donné par `sort()`. Vous pouvez habituellement doubler les boucles pour obtenir le même effet, car la boucle intérieure est considérée comme un bloc qui boucle un seule fois. Voir `last()` dans *perlfunc*.

Can't "next" outside a block

(F) L'état « `next` » a été exécuté pour réitérer le bloc courant, mais il n'y a pas de bloc courant. Remarquez que les blocs « `if` » ou « `else` » ne comptent pas comme des blocs « `loop` », ou comme un bloc donné par `sort()`. Vous pouvez habituellement doubler les boucles pour obtenir le même effet, car la boucle intérieure est considérée comme un bloc qui boucle un seule fois. Voir `next()` dans *perlfunc*.

Can't "redo" outside a block

(F) L'état « `redo` » a été exécuté pour recommencer le bloc courant, mais il n'y a pas de bloc courant. Remarquez que les blocs « `if` » ou « `else` » ne comptent pas comme des blocs « `loop` », ou comme un bloc donné par `sort()`. Vous pouvez habituellement doubler les boucles pour obtenir le même effet, car la boucle intérieure est considérée comme un bloc qui boucle un seule fois. Voir `redo()` dans *perlfunc*.

Can't bless non-reference value

(F) Seules les références en dur peuvent être consacrées. C'est comme cela que Perl « renforce » l'encapsulation des objets. Voir *perlobj*.

Can't break at that line

(S) C'est un message d'erreur qui est seulement affiché pendant l'exécution avec le débogueur, indiquant que la ligne spécifiée n'est pas l'emplacement d'une expression qui peut être stoppée.

Can't call method "%s" in empty package "%s"

(F) Vous appelez une méthode correctement, et vous avez indiqué correctement le paquetage fonctionnant comme une classe, mais ce paquetage n'a RIEN de défini, et n'a pas de méthodes. Voir *perlobj*.

Can't call method "%s" on unblest reference

(F) Un appel de méthode doit savoir dans quel paquetage il est supposé être lancé. Cela se trouve habituellement dans la référence de l'objet que vous soumettez, mais vous ne soumettez pas de référence d'objet dans ce cas là. Une référence n'est pas une référence d'objet jusqu'à qu'il ait été consacré (blessed). Voir *perlobj*.

Can't call method "%s" without a package or object reference

(F) Vous utilisez la syntaxe d'un appel de méthode, mais l'endroit rempli par la référence d'objet ou le nom du paquetage contient une expression qui retourne une valeur définie qui n'est ni une référence d'objet ni un nom de paquetage. Quelque chose comme ça reproduit l'erreur :

```
$BADREF = 42;
process $BADREF 1, 2, 3;
$BADREF->process(1, 2, 3);
```

Can't call method "%s" on an undefined value

(F) Vous utilisez la syntaxe d'un appel de méthode, mais l'endroit rempli par la référence d'objet ou le nom du paquetage contient une valeur indéfinie. Quelque chose comme ça reproduit l'erreur :

```
$BADREF = undef;
process $BADREF 1, 2, 3;
$BADREF->process(1, 2, 3);
```

Can't chdir to %s

(F) Vous appelez `perl -x/foo/bar`, mais `/foo/bar` n'est pas un répertoire dans lequel vous pouvez entrer, probablement parce qu'il n'existe pas.

Can't coerce %s to integer in %s

(F) Certain types de SVs, en particulier la table des entrées des symboles réels (typeglobs), ne peuvent être forcés ou stoppés d'être ce qu'ils sont. Donc vous ne pouvez pas faire quelque chose comme ça :

```
*foo += 1;
```

Vous POUVEZ dire

```
$foo = *foo;
$foo += 1;
```

mais alors `$foo` ne contient plus de glob.

Can't coerce %s to number in %s

(F) Certain types de SVs, en particulier la table des entrées des symboles réels (typeglobs), ne peuvent être forcés ou stoppés d'être ce qu'ils sont.

Can't coerce %s to string in %s

(F) Certain types de SVs, en particulier la table des entrées des symboles réels (typeglobs), ne peuvent être forcés ou stoppés d'être ce qu'ils sont.

Can't coerce array into hash

(F) Vous utilisez un tableau là où un tableau associatif est attendu, mais le tableau n'a pas d'informations sur comment passer des clés aux indices de tableau. Vous ne pouvez faire cela qu'avec les tableaux dont leurs références associatives sont d'index 0.

Can't create pipe mailbox

(P) Une erreur spécifique à VMS. Le `process(???)` souffre de quotas dépassés ou d'autres problèmes de limitation.

Can't declare %s in my

(F) Seuls les scalaires, les tableaux et les tableaux associatifs peuvent être déclarés comme variables lexicales. Ils doivent avoir un identifiant ordinaire comme nom.

Can't do inplace edit on %s: %s

(S) La création du nouveau fichier a échoué à cause la raison indiquée.

Can't do inplace edit without backup

(F) Vous êtes sur un système comme MS-DOS qui s'embrouille si il essaie de lire un fichier supprimé (mais toujours ouvert). Vous devez dire `-i.bak`, ou quelque chose d'identique.

Can't do inplace edit: %s > 14 characters

(S) Il n'y a pas assez de place dans le nom de fichier pour faire un backup de ce fichier.

Can't do inplace edit: %s is not a regular file

(S) Vous essayez d'utiliser l'option `-i` sur un fichier spécial, comme un fichier de `/dev` ou un FIFO. Le fichier est ignoré.

Can't do setegid!

(P) L'appel à `setegid()` a échoué pour certaines raisons dans l'émulateur `setuid` de `suidperl`.

Can't do seteuid!

(P) L'émulateur `setuid` de `suidperl` a échoué pour certaines raisons.

Can't do setuid

(F) Cela veut typiquement dire que le perl ordinaire essaye d'exécuter `suidperl` pour faire une émulation `setuid`, mais ne peut le faire. Il cherche un nom de la forme `perl5.000` dans le même répertoire que réside l'exécutable `perl`, typiquement sous `/usr/local/bin` sur les machines Unix. Si le fichier s'y trouve, vérifiez les permissions d'exécution. Si ce n'est pas cela, demandez à votre administrateur système pourquoi il l'a désactivé.

Can't do waitpid with flags

(F) Cette machine n'a pas ni `waitpid()` ni `wait4()`, donc seul `waitpid()` sans aucun paramètre est émulé (???)

Can't do {n,m} with n > m

(F) Le minimum doit être inférieur ou égal au maximum. Si vous voulez vraiment que votre expression rationnelle ne trouve rien, faites juste `{0}`. Voir *perlre*.

Can't emulate -%s on #! line

(F) La ligne `#!` spécifie une option qui n'a pas de sens à ce point. Par exemple, il serait idiot de mettre l'option `-x` sur cette ligne `#!`.

Can't exec "%s": %s

(W) L'appel à la fonction `system()`, `exec()`, ou un `open()` sur un pipe n'a pu s'exécuter pour la raison indiquée. Les raisons typiques sont : les permissions sur le fichier sont mauvaises, le fichier n'a pu être trouvé dans `$ENV{PATH}`, l'exécutable en question a été compilé sur une autre architecture, ou la ligne `#!` pointe vers un interpréteur qui ne peut pas être lancé pour une raison similaire. (Ou peut-être votre système ne supporte pas `#!` après tout).

Can't exec %s

(F) Perl essaye d'exécuter le programme indiqué car c'est ce qui est spécifié à la ligne `#!`. Si ce n'est pas ce que vous voulez, vous devrez mettre « `perl` » quelque part sur la première ligne.

Can't execute %s

(F) Vous utilisez l'option `-S`, mais la copie du script à exécuter trouvée dans le `PATH` n'a pas les bonnes permissions.

Can't find %s on PATH, '.' not in PATH

(F) Vous utilisez l'option `-S`, mais le script à exécuter ne peut être trouvé dans le `PATH`, ou avec des permissions incorrectes. Le script existe dans le répertoire courant, mais `PATH` l'empêche de se lancer.

Can't find %s on PATH

(F) Vous utilisez l'option `-S`, mais le script à exécuter ne peut être trouvé dans le `PATH`.

Can't find label %s

(F) Vous dites par un « `goto` » d'aller à une étiquette qui n'est mentionnée nulle part, ou a un endroit qu'il n'est possible de joindre. Voir `goto()` dans *perlfunc*.

Can't find string terminator %s anywhere before EOF

(F) Les chaînes de caractères en Perl peuvent s'étirer sur plusieurs lignes. Ce message veut dire que vous avez oublié le délimitateur fermant l'expression. Comme les parenthèses protégées par des quotes comptent comme un niveau, il manque dans l'exemple suivant la parenthèse finale :

```
print q(The character '(' starts a side comment.);
```

Si vous obtenez cela depuis un « here-document », vous devez avoir mis des espaces blancs non visibles avant ou après la marque de fermeture. Un bon éditeur de programmeur doit pouvoir vous aider à les trouver.

Can't fork

(F) Une erreur fatale est survenue alors lors de l'essai d'appel à `fork` lorsqu'il a ouvert un pipe.

Can't get filespec - stale stat buffer?

(S) Une alerte spécifique à VMS. Cela est causé par les différences entre les contrôles d'accès sous VMS et sous le modèle Unix. Sous VMS, les contrôles d'accès sont faits par le nom de fichier, plutôt que par bits dans le tampon de stat, alors que les ACL et autres protections peuvent être pris dans le compte. Malheureusement, Perl suppose que le tampon de stat contient toutes les informations nécessaires et le fournit, au lieu de la spécification du fichier, à la fonction de contrôle d'accès. Il va essayer de retirer les spécifications du fichier en utilisant le nom de device et de FID présent dans le tampon de stat, mais cela ne marchera que si vous avez fait un appel subséquent à la fonction `CTRL stat()`, car le nom de device est écrasé à chaque appel. Si cette alerte apparaît, la recherche de nom échoue, et la fonction de contrôle d'accès rend la main et retourne `FALSE`, juste pour être conservateur. (Remarque : la fonction de contrôle d'accès connaît l'opérateur `stat` de Perl et les tests de fichiers, donc vous ne devriez jamais voir ce message en réponse à une commande Perl ; cela arrive seulement si certains codes internes prennent les tampons de stat à la légère.)

Can't get pipe mailbox device name

(P) Une erreur spécifique à VMS. Après avoir créé une boîte aux lettres pour agir en tant que pipe, Perl ne peut retirer son nom pour un usage ultérieur.

Can't get SYSGEN parameter value for MAXBUF

(P) Une erreur spécifique à VMS. Perl a demandé à `$GETSYI` quelle taille vous vouliez pour vos boîtes aux lettres, et n'a pas obtenu de réponse.

Can't goto subroutine outside a subroutine

(F) L'appel à la très magique « `goto` subroutine » peut seulement remplacer l'appel d'une fonction par une autre. Il ne peut fabriquer one out of whole cloth(???). En général, elle ne peut être appelé que depuis une fonction d'`AUTOLOAD`. Voir `goto()` dans *perlfunc*.

Can't goto subroutine from an eval-string

(F) L'appel à « `goto` subroutine » ne peut être utilisé pour sortir d'une chaîne utilisée par `eval()`. (Vous pouvez l'utiliser pour sortir d'un `eval() {BLOCK}`, mais ce n'est sûrement pas ce que vous voulez.

Can't localize through a reference

(F) Vous dites quelque chose comme `local $$ref`, ce que Perl ne peut accepter, car quand il va restaurer l'ancienne valeur de ce que `$ref` référençait, il ne peut être sûr que `$ref` est toujours une référence.

Can't localize lexical variable %s

(F) Vous utilisez `local` sur une variable qui a déjà été déclarée auparavant comme une variable lexicale à l'aide de « `my` ». Cela n'est pas permis. Si vous voulez localiser une variable de paquetage du même nom, qualifiez-la avec le nom du paquetage.

Can't localize pseudo-hash element

(F) Vous dites quelque chose comme `local $ar->{'key'}`, où `$ar` est une référence vers un pseudo-hash. Cela n'a pas été implémenté pour le moment, mais vous pouvez obtenir un effet similaire en localisant l'élément de tableau correspondant par `local $ar->[$ar->[0]{'key'}]`.

Can't locate auto/%s.al in @INC

(F) Une fonction (ou une méthode) a été appelé dans un paquetage qui autorise le chargement automatique, mais il n'y a pas de fonction à charger automatiquement. Le plus probable est une faute de frappe sur le nom de fonction/méthode ou une erreur pour utiliser `AutoSplit` sur le fichier, entraîné par `make install`.

Can't locate %s in @INC

(F) Vous dites d'exécuter (`run`) (ou `require`, ou `use`) un fichier qui ne peut être trouvé dans aucune des librairies mentionnées dans `@INC`. Peut-être devriez-vous positionner la variable d'environnement `PERL5LIB` ou `PERL5OPT` pour dire où se trouve votre librairie supplémentaire, ou peut-être le script nécessite que vous ajoutiez le nom de votre librairie à `@INC`. Ou peut-être avez-vous mal épilé le nom de votre fichier. Voir `require()` dans *perlfunc*.

Can't locate object method "%s" via package "%s"

(F) Vous appelez un méthode correctement, et vous avez indiqué correctement un paquetage fonctionnant comme une classe, mais le paquetage ne définit pas cette méthode, ni aucune de ses classes parentes. Voir *perlobj*.

Can't locate package %s for @%s::ISA

(W) Le tableau `@ISA` contient un nom d'un autre paquetage qu'il ne semble pas exister.

Can't make list assignment to \%ENV on this system

(F) L'affectation de listes sur `%ENV` n'est pas supportée sur certains systèmes, notamment VMS.

Can't modify %s in %s

(F) Vous n'êtes pas autorisé à faire d'affectation sur l'élément indiqué, ou autrement essayez de le changer, comme avec une incrémentation automatique.

Can't modify nonexistent substring

(P) La fonction interne qui fait l'affectation à `substr()` a capturé un `NULL`.

Can't msgrcv to read-only var

(F) La cible de `msgrcv` doit être modifiable pour être utilisée comme un buffer reçu.

Can't open %s: %s

(S) L'ouverture implicite d'un fichier en utilisant `<>` sur le descripteur de fichier, implicitement via les options `-n` ou `-p` en ligne de commandes, ou explicitement, a échoué à cause de la raison indiquée. Habituellement c'est parce que vous n'avez pas les permissions de lecture sur le fichier que vous avez nommé sur la ligne de commande.

Can't open bidirectional pipe

(W) Vous essayez de dire `open (CMD, "|cmd|")`, ce qui n'est pas supporté. Vous pouvez essayer un des nombreux modules de la librairie Perl pour faire cela, comme `IPC::Open2`. Alternativement, dirigez la sortie du pipe dans un fichier en utilisant `<>`, et ensuite lisez-le dans un descripteur de fichier différent.

Can't open error file %s as stderr

(F) Une erreur spécifique à VMS. Perl fait ses propres redirections de ligne de commande, et ne peut ouvrir le fichier spécifié après `'2>'` ou `'2>>'` sur la ligne de commande pour y écrire.

Can't open input file %s as stdin

(F) Une erreur spécifique à VMS. Perl fait ses propres redirections de ligne de commande, et ne peut ouvrir le fichier spécifié après `'<'` sur la ligne de commande pour le lire.

Can't open output file %s as stdout

(F) Une erreur spécifique à VMS. Perl fait ses propres redirections de ligne de commande, et ne peut ouvrir le fichier spécifié après `'>'` ou `'>>'` sur la ligne de commande pour y écrire.

Can't open output pipe (name: %s)

(P) Une erreur spécifique à VMS. Perl fait ses propres redirections de ligne de commande, et ne peut ouvrir le pipe destiné à la sortie des données.

Can't open perl script "%s": %s

(F) Le script spécifié ne peut être ouvert pour la raison indiquée.

Can't redefine active sort subroutine %s

(F) Perl optimise l'appel interne aux sous-programmes `sort` et garde des pointeurs vers ceux-ci. Vous essayez de redéfinir un des sous-programmes qui est actuellement actif, ce qui n'est pas permis. Si vous voulez vraiment faire cela, vous pouvez écrire `sort { &func } @x` au lieu de `sort func @x`.

Can't rename %s to %s: %s, skipping file

(S) L'opération pour renommé effectué par l'option `-i` a échoué pour une raison, probablement parce que vous n'avez pas le droit d'écrire dans le répertoire.

Can't reopen input pipe (name: %s) in binary mode

(P) Une erreur spécifique à VMS. Perl pense que `stdin` est un pipe, et essaye de le rouvrir pour le faire accepter des données. Hélas, cela échoue.

Can't reswap uid and euid

(P) L'appel à la fonction `setreuid()` a échoué pour un raison dans l'émulateur `setuid` de `suidperl`.

Can't return outside a subroutine

(F) L'expression `return` a été exécutée dans la ligne principale, où il n'y a aucun appel de sous-programme à retourner. Voir *perlsub*.

Can't stat script "%s"

(P) Pour certaines raisons, vous ne pouvez faire un `fstat()` sur le script tant que vous l'avez déjà ouvert. Bizarre.

Can't swap uid and euid

(P) L'appel à la fonction `setreuid()` a échoué pour un raison dans l'émulateur `setuid` de `suidperl`.

Can't take log of %g

(F) Pour les nombres réels ordinaires, vous ne pouvez faire le logarithme d'un nombre négatif ou nul. Il existe un paquetage `Math::Complex` en standard avec Perl, si c'est vraiment pour faire cela sur un nombre négatif.

Can't take sqrt of %g

(F) Pour les nombres réels ordinaires, vous ne pouvez obtenir la racine carrée d'un nombre négatif. Il existe le paquetage `Math::Complex` en standard avec Perl, qui lui, si vous le voulez vraiment, peut faire cela.

Can't undef active subroutine

(F) Vous ne pouvez effacer une fonction qui est utilisée actuellement.

Vous pouvez cependant, la redéfinir pendant qu'elle tourne, et faire un undef sur la fonction redéfinie pendant que l'ancienne fonction tourne. Go figure(???)

Can't unshift

(F) Vous essayez de faire un unshift sur un tableau « irréal » qui ne peut accepter cette opération, comme la pile principale de Perl.

Can't upgrade that kind of scalar

(P) La fonction interne sv_upgrade a ajouté un « membre » à un SV, ce qui fait qu'il se trouve dans une sorte de SV plus spécialisée. Les différents premiers types de SV sont si spécialisés, cependant, qu'ils ne peuvent être inter-convertis. Ce message indique qu'une telle conversion a été tentée.

Can't upgrade to undef

(P) La non définie SV est le bas du « totem pole », dans le projet des « upgradability ». Arriver à undef indique une(???) dans le code qui appelle sv_upgrade.

Can't use %%! because Errno.pm is not available

(F) La première fois que le tableau associatif %! est utilisé, perl automatiquement charge le module Errno.pm. Le module Errno est attendu pour attacher au tableau %! les noms symboliques pour les valeurs des numéros d'erreur de \$!.

Can't use "my %s" in sort comparison

(F) Les variables globales \$a et \$b sont réservés pour les comparaisons avec sort. Vous mentionnez \$a ou \$b dans la même ligne que l'opérateur <=> ou cmp, et la variable qui a été déclarée plus tôt comme variable lexicale. Ou qualifiez la variable de sort avec le nom de paquetage, ou renommez la variable lexicale.

Can't use %s for loop variable

(F) Seulement une variable scalaire simple peut être utilisée comme variable de boucle dans un foreach.

Can't use %s ref as %s ref

(F) Vous avez mélangé vos types de références. Vous avez déréférencé la référence du type nécessaire. Vous pouvez utiliser la fonction ref() pour tester le type de la référence, si nécessaire.

Can't use \1 to mean \$1 in expression

(W) Dans les expressions ordinaires, la barre oblique inverse est un opérateur unaire qui crée une référence vers cet argument. L'utilisation de barre oblique inverse pour indiquer une backreference sur une sous-chaîne correspondante est valide seulement comme partie du motif de l'expression rationnelle. Essayer de faire ça dans un code ordinaire Perl produit une valeur qui ressemble à SCALAR(0xdecaf). Utilisez l'élément \$1 à la place.

Can't use bareword ("%s") as %s ref while "strict refs" in use

(F) Les références en dur sont les seules autorisées par « strict refs ». Les références symboliques sont interdites. Voir *perlref*.

Can't use string ("%s") as %s ref while "strict refs" in use

(F) Seules les affectations « en dur » sont permises par « strict refs ». Les références symboliques sont interdites. Voir *perlref*.

Can't use an undefined value as %s reference

(F) Une valeur utilisée comme une référence en dur ou comme une référence symbolique doit avoir une valeur définie. Cela aide à débloquer certaines erreurs insidieuses.

Can't use global %s in "my"

(F) Vous essayez de déclarer une variable magique en tant que variable lexicale. Cela n'est pas permis, car la magique peut être lié qu'à un seul endroit (nommé variable globale) et il serait incroyablement confus d'avoir des variables dans votre programme qui ressemblent à une variable magique mais qui n'en sont pas une.

Can't use subscript on %s

(F) Le compilateur essaye d'interpréter une expression entre parenthèses comme un sous-script. Mais à gauche des parenthèses il y a une expression qui ne ressemble pas à un tableau de références, ni quoi que ce soit que l'on puisse interpréter comme un sous-script.

Can't x= to read-only value

(F) Vous essayez de répéter une valeur constante (souvent une valeur indéfinie) avec un opérateur d'affectation, ce qui implique de modifier la valeur elle-même. Peut-être que vous devriez copier la valeur dans un endroit temporaire, et recommencer.

Cannot find an opnumber for "%s"

(F) Une chaîne de la forme `CORE::word` a été donné à `prototype()`, mais il n'existe pas de variable interne nommée `word`.

Cannot resolve method '%s' overloading '%s' in package '%s'

(F|P) Une erreur lors de la résolution de la surcharge spécifiée par le nom de méthode (à l'opposé d'une référence de sous-fonction) : aucune méthode de ce nom là peut être appelé via ce paquetage. Si le nom de la méthode est ???, il s'agit d'une erreur interne.

Character class syntax [.] is reserved for future extensions

(W) À l'intérieur des classes de caractères dans les expressions rationnelles ([]) la syntaxe commençant par « [. » et se terminant par « .] » est réservé pour les extensions futures. Si vous devez représenter cette séquence de caractères dans une classe de caractères dans une expression rationnelle, cotez simplement les crochets avec une barre oblique inverse : « \[. » et « \.] ».

Character class syntax [:] is reserved for future extensions

(W) À l'intérieur des classes de caractères dans les expressions rationnelles ([]) la syntaxe commençant par « [: » et se terminant par « :] » est réservé pour les extensions futures. Si vous devez représenter cette séquence de caractères dans une classe de caractères dans une expression rationnelle, cotez simplement les crochets avec une barre oblique inverse : « \[: » et « \:] ».

Character class syntax [=] is reserved for future extensions

(W) À l'intérieur des classes de caractères dans les expressions rationnelles ([]) la syntaxe commençant par « [= » et se terminant par « =] » est réservé pour les extensions futures. Si vous devez représenter cette séquence de caractères dans une classe de caractères dans une expression rationnelle, cotez simplement les crochets avec une barre oblique inverse : « \[= » et « =\] ».

chmod: mode argument is missing initial

(W) Un utilisateur novice aura tendance à faire

```
chmod 777, $filename
```

en ne réalisant pas que `777` est interprété comme un nombre décimal, équivalent à `01411`. Les constantes octales sont introduites en utilisant un `0` en Perl, comme en C.

Close on unopened file <%s>

(W) Vous essayez de fermer un descripteur de fichier qui n'a jamais été ouvert.

Compilation failed in require

(F) Perl ne peut compiler le fichier spécifié à l'aide la directive `require`. Perl utilise ce message générique quand aucune des erreurs rencontrées n'est assez grave pour stopper la compilation immédiatement.

Complex regular subexpression recursion limit (%d) exceeded

(W) Le moteur d'expressions rationnelles utilise la récursion dans les situations complexe où le back-tracking est nécessaire. La profondeur de la récursion est limitée à `32766`, ou peut-être moins sur les architectures où la pile ne peut grossir arbitrairement. (Les situations « simples » et « moyennes » peuvent être capturées sans récursion et ne sont pas sujet à une limite.) Essayer de raccourcir la chaîne examinée, et boucler sur le code Perl (ex : avec `while` plutôt que par le moteur d'expressions rationnelles ; ou réécrivez votre expression rationnelle pour qu'elle soit plus simple ou moins référencée. (Voir *perlbook* pour des informations sur *Mastering Regular Expressions*.)

connect() on closed fd

(W) Vous essayez de faire un `connect` sur une socket fermée. Peut-être avez-vous oublié de vérifier la valeur retournée par l'appel à `socket()` ? Voir `connect()` dans *perlfunc*.

Constant subroutine %s redefined

(S) Vous redéfinissez une fonction qui a été marquée auparavant comme `inlining`. Voir Fonctions Constantes in *perlsub* pour les commentaires et les issues.

Constant subroutine %s undefined

(S) Vous supprimez une fonction qui a été marquée auparavant comme `inlining`. Voir Constant Functions in *perlsub* pour les commentaires et les issues.

Copy method did not return a reference

(F) La méthode qui surclasse « = » est buggée. Voir Copy Constructor in *overload*.

Corrupt malloc ptr 0x%lx at 0x%lx

(P) Le paquetage `malloc` distribué avec Perl a eu une erreur interne.

corrupted regexp pointers

(P) Le moteur d'expressions rationnelles a été abusé par l'expression rationnelle que vous lui avez fournie.

corrupted regexp program

(P) Le moteur d'expressions rationnelles a analysé une regexp sans un « magic number » valide.

Deep recursion on subroutine "%s"

(W) Ce sous-programme s'est appelé lui-même (directement ou indirectement) 100 fois plus de fois qu'il n'a retourné une valeur. Cela indique probablement une récursion infinie, à moins que vous n'écriviez un étrange programme de benchmark, dans ce cas cela indique quelque chose d'autre.

Delimiter for here document is too long

(F) Dans un « here document » construit comme <<FOO, l'étiquette FOO est trop long pour que Perl puisse la traiter. Vous devez être sérieusement tordu pour écrire un code qui entraîne cette erreur.

Did you mean & %s instead?

(W) Vous faites probablement référence à une sous-fonction importée &FOO en faisant \$FOO ou quelque chose du genre.

Did you mean \$ or @ instead of %?

(W) Vous avez probablement dit %hash{\$key} alors que vous pensiez \$hash{\$key} ou @hash{@keys}. En d'autres termes, peut-être pensez-vous à %hash et vous avez et ca l'a emporté.???

Died

(F) Vous appelez die() avec une chaîne vide (l'équivalent de die "") ou vous l'appellez sans arguments et avec \$@ et \$_ vides.

Do you need to predeclare %s?

(S) C'est un message donné à titre indicatif en conjonction avec le message « %s found where operator expected ». Cela veut souvent dire qu'un nom de fonction ou de module est référencé alors qu'il n'est pas encore défini pour le moment. C'est peut être un problème d'ordre dans votre fichier, ou parce qu'il manque un « sub », « paquetage », « require », ou « use ». Si vous référencez quelque chose qui n'est pas encore défini pour le moment, vous n'avez pas à définir la fonction ou le paquetage avant cet endroit. Vous pouvez utiliser « un sub foo; » ou un paquetage « FOO; » vide pour entrer une déclaration « devant ».

Don't know how to handle magic of type '%s'

(P) Le traitement interne des variables magiques a été endommagé.

do_study: out of memory

(P) Ce message peut être capturé par l'appel de safemalloc().

Duplicate free() ignored

(S) Une fonction interne a appelé free() sur quelque chose qui a déjà été libéré.

elseif should be elsif

(S) Il n'y a pas de mot-clef « elseif » en Perl car Larry pense que c'est très laid. Votre code sera interprété comme un essai d'appel à la méthode nommée « elseif » pour la classe retournée par le bloc suivant. Ce n'est sûrement pas ce que vous voulez.

END failed—cleanup aborted

(F) Une exception non capturable a été levée pendant que la sous-fonction END était exécuté. L'interpréteur est sorti immédiatement.

Error converting file specification %s

(F) Une erreur spécifique à VMS. Comme Perl doit traiter avec des spécifications de fichiers autres que la syntaxe VMS ou Unix, il convertit ceux-ci dans un format unique où il peut opérer avec directement. Ou vous avez passé des spécifications de fichier non valide, ou vous avez trouvé un cas où la routine de conversion ne peut rien faire. Drat.???

%s: Eval-group in insecure regular expression

(F) Perl a détecté des données souillées quand il essaye de compiler une expression rationnelle qui contient le (?{ ... }) d'affectation de longueur zéro, ce qui n'est pas sécurisé. Voir (?{ code }) in *perlre*, and *perlsec*.

%s: Eval-group not allowed, use re 'eval'

(F) Une expression rationnelle contient une longueur nulle dans (?{ ... }), mais cette construction est seulement permise quand le code use re 'eval' est en action. Voir (?{ code }) in *perlre*.

%s: Eval-group not allowed at run time

(F) Perl essaye de compiler une expression rationnelle qui contient (?{ ... }) affectation de longueur nulle à l'exécution, comme si l'expression contenait les valeurs interpolées. Comme c'est un risque de sécurité, cela n'est pas permis. Si vous insistez, vous devez le faire en construisant explicitement votre expression depuis une chaîne interpolée à l'exécution et l'utiliser dans un eval().

Excessively long <> operator

(F) Le contenu des opérateurs <> ne doit pas excéder la taille maximum d'un identifiant Perl. Si vous essayez juste d'obtenir les extensions d'une longue liste de fichiers, essayez d'utiliser l'opérateur glob(), ou mettez le noms de fichiers dans une variable et faites un glob() dessus.

Execution of %s aborted due to compilation errors

(F) Le message final lorsqu'une compilation Perl échoue.

Exiting eval via %s

(W) Vous êtes sorti de Perl d'une façon non conventionnelle, comme un goto, ou un contrôle de boucle.

Exiting pseudo-block via %s

(W) Vous êtes sorti d'un bloc spécial de constructeur (comme un bloc sort ou une fonction) d'une façon non conventionnelle, comme un goto, ou un contrôle de boucle. Voir sort () dans *perlfunc*.

Exiting subroutine via %s

(W) Vous êtes sorti d'une fonction d'une façon non conventionnelle, comme un goto, ou un contrôle de boucle.

Exiting substitution via %s

(W) Vous êtes sorti d'une substitution d'une façon non conventionnelle, comme un goto, un return, ou un contrôle de boucle.

Explicit blessing to '' (assuming package main)

(W) Vous consacrez (to bless) une référence à une chaîne de longueur nulle. Cela a pour effet de consacrer la référence dans le paquetage principal. Ce n'est pas habituellement ce que vous voulez. Fournissez un paquetage-cible par défaut, ex bless(\$ref, \$p or 'MyPaquetage');

Fatal VMS error at %s, line %d

(P) Erreur spécifique à VMS. Il s'est passé quelque chose dans un service du système VMS ou d'une fonction RTL ; L'état de Perl à sa sortie peut fournir plus de détails. Le nom de fichier dans « at %s » et le numéro de ligne dans « line %d » vous indique quelle section du code source Perl est gêné.

fcntl is not implemented

(F) Votre machine apparemment n'implémente pas fcntl(). C'est quoi, un PDP-11 ou quelque chose de similaire ?

Filehandle %s never opened

(W) Une opération d'E/S a été tentée sur un descripteur de fichier qui n'a jamais été initialisé. Vous devez faire un appel à open() ou à socket(), ou appelez un constructeur depuis le paquetage FileHandle.

Filehandle %s opened for only input

(W) Vous tentez d'écrire dans un descripteur de fichier ouvert en lecture seulement. Si vous désirez que ce soit un descripteur de fichier ouvert en écriture, vous devez l'ouvrir avec « +< » ou « \$+> » ou « +>> » au lieu de « < » ou rien du tout. Si vous désirez uniquement écrire dans le fichier, utilisez « > » ou ">>". Voir open () dans *perlfunc*.

Filehandle opened for only input

(W) Vous tentez d'écrire dans un descripteur de fichier ouvert en lecture seulement. Si vous désirez que ce soit un descripteur de fichier ouvert en écriture, vous devez l'ouvrir avec « +< » ou « +> » ou « +>> » au lieu de « < » ou rien du tout. Si vous désirez uniquement écrire dans le fichier, utilisez « > » ou ">>". Voir open () dans *perlfunc*.

Final \$ should be \ \$ or \$name

(F) Vous devez maintenant décider si le \$ final dans une chaîne doit être interprétée comme le signe littéraire dollar, ou doit être interprétée comme l'introduction d'un nom de variable qui apparaît comme manquant. Donc vous devez ou mettre la barre oblique inverse ou bien le nom.

Final @ should be \@ or @name

(F) Vous devez maintenant décider si le @ dans une chaîne doit être interprété comme le signe littéral « at », ou si c'est dans le but d'introduire un nom de variable qui apparaît comme manquant. Donc vous devez mettre soit une barre oblique inverse soit le nom.

Format %s redefined

(W) Vous redéfinissez un format. Pour supprimer ce message faire

```
{
    local $^W = 0;
    eval "format NAME =...";
}
```

Format not terminated

(F) Un format doit être terminé par une ligne avec un point uniquement. Perl est arrivé en fin de fichier sans trouver une telle ligne.

Found = in conditional, should be ==

(W) Vous dites

```
if ($foo = 123)
```

alors que vous pensez

```
if ($foo == 123)
```

(ou quelque chose de similaire).

gdbm store returned %d, errno %d, key "%s"

(S) Un message d'alerte de l'extension GDBM_File qui a échoué un stockage.

gethostent not implemented

(F) Votre librairie C n'implémente apparemment pas `gethostent()`, probablement parce que si elle le fait, il se sentirait moralement obligé de rendre chaque hostname d'Internet.

get{sock,peer}name() on closed fd

(W) Vous essayez d'obtenir une socket ou une socket parente d'une socket fermée. Peut-être avez-vous oublié de vérifier la valeur retournée par l'appel à `socket()` ?

getpwnam returned invalid UIC %#o for user "%s"

(S) Message spécifique à VMS. L'appel à `sys$getuai` souligne l'opérateur `getpwnam` qui a retourné une UIC non valide.

Glob not terminated

(F) L'analyseur syntaxique a vu une parenthèse gauche à la place de ce qu'il attendait être un terme, donc il recherche la parenthèse droite correspondante, et ne la trouve pas. Il y a des chances que vous ayez oublié des parenthèses nécessaires plus tôt dans la ligne.

Global symbol "%s" requires explicit package name

(F) Vous spécifiez « `use strict vars` », ce qui indique que toutes les variables doivent être ou déclarées locales (en utilisant « `my` »), ou explicitement qualifiées pour dire dans quel paquetage la variable globale est déclarée (en utilisant « `::` »).

goto must have label

(F) Au contraire de `next` et de `last`, vous n'êtes pas autorisé à aller à une destination non spécifiée. Voir `goto()` dans *perlfunc*.

Had to create %s unexpectedly

(S) Une fonction demande un symbole depuis la table des symboles où celle-ci doit déjà exister, mais pour une raison, elle n'existe pas, et a été créée en urgence pour éviter un core dump.

Hash % %s missing the % in argument %d of %s()

(D) Seuls les vraiment vieux Perl vous permettaient d'omettre le `%` dans un nom de tableau associatif à certains endroits. C'est maintenant largement obsolète.

Identifiant too long

(F) Perl limite la taille des identifiants (noms des variables, fonctions, etc.) à 250 caractères pour les noms simples, et un peu plus pour les noms composés (comme `$A:B`). Vous avez excédé les limites de Perl. Les versions futures de Perl vont éliminer cette limitation arbitraire.

Ill-formed logical name |%s| in prime_env_iter

(W) Une alerte spécifique à VMS. Un nom logique a été rencontré lors de la préparation d'itération sur `%ENV`, ce qui viole les règles syntaxiques gouvernant les noms logiques. Comme ils ne peuvent être transmis normalement, ils sont sautés, et n'apparaissent pas dans `%ENV`. Cela peut-être un événement bénin, comme dans certains paquetages de software qui peuvent directement modifier le nom logique des tables et introduire des noms non standards, ou cela peut indiquer que le nom logique a été corrompu.

Illegal character %s (carriage return)

(F) Le caractère 'carriage return' a été trouvé en entrée. C'est une erreur, pas une alerte, car le carriage return peut casser une chaîne multi-ligne, incluant le document (e.g., `print <<EOF;`).

Sous Unix, cette erreur est habituellement causée par l'exécution de code Perl – ou par le programme principal, un module, ou un `eval` sur une chaîne – qui a été transféré à travers le réseau depuis un système non-Unix qui ne convertit pas proprement le format des fichiers textes.

Sous certains systèmes qui utilisent quelque chose d'autre que le `'\n'` pour délimiter les lignes d'un texte, cette erreur peut être causée par la lecture de code Perl depuis un descripteur de fichier qui est en mode binaire (comme positionné par l'opérateur `binmode`).

Dans certains cas, le code Perl en question doit probablement être converti avec quelque chose comme `s/\x0D\x0A?/\n/g` avant de pouvoir être exécuté.

Illegal division by zero

(F) Vous essayez de diviser un nombre par 0. Ou quelque chose n'est pas bon dans votre logique, ou vous devez placer une condition pour vous garder de cette entrée dénuée de sens.

Illegal modulus zero

(F) Vous essayez de diviser un nombre par 0 pour obtenir le reste. La plupart des nombres ne peuvent faire cela facilement.

Illegal octal digit

(F) Vous avez utilisé un 8 ou un 9 dans un nombre octal.

Illegal octal digit ignored

(W) Vous avez essayé d'utiliser un 8 ou un 9 dans un nombre octal. L'interprétation du nombre octal est stoppé avant le 8 ou le 9.

Illegal hex digit ignored

(W) Vous avez essayé d'utiliser un caractère autre que 0 - 9 ou A - F dans un nombre hexadécimal. L'interprétation du nombre hexadécimal est stoppée avant le caractère illégal.

Illegal switch in PERL5OPT: %s

(X) La variable d'environnement PERL5OPT peut être utilisée seulement pour positionner les options suivantes : **-[DIMUdmw]**.

In string, @ %s now must be written as \@ %s

(F) Cela est utilisé pour indiquer sur Perl ??? essaie de deviner si vous voulez un tableau interpolé ou un littéral @. Cela arrive quand la chaîne est utilisée la première fois à l'exécution. Maintenant les chaînes sont parsées au moment de la compilation, et les instances ambiguës de @ doivent être éclaircies, soit en mettant une barre oblique inverse pour indiquer qu'il s'agit d'un littéral, ou en déclarant (ou en utilisant) le tableau dans le programme avant la chaîne (lexicalement). (Parfois cela veut simplement dire qu'un @ peut être interprété comme un tableau.)

Insecure dependency in %s

(F) Vous essayez de faire quelque chose que le mécanisme de sécurité sur l'entâchement des variables n'apprécie pas. Ce mécanisme est activé quand vous exécutez un script setuid ou setgid, ou si vous spécifiez l'option **-T** pour l'activer explicitement. Le mécanisme de sécurité sur l'entâchement des variables marque tous les données dérivées directement ou indirectement de l'utilisateur, qui n'est pas considéré comme allié dans vos transactions. Si une de ces données est utilisée dans une opération « dangereuse », vous obtenez cette erreur. Voir *perlsec* pour plus d'informations.

Insecure directory in %s

(F) Vous ne pouvez utiliser `system()`, `exec()`, ou ouvrir un pipe dans un script setuid ou setgid si `$ENV{PATH}` contient un répertoire qui a les droits en écriture pour tout le monde. Voir *perlsec*.

Insecure \$ENV{ %s} while running %s

(F) Vous ne pouvez utiliser `system()`, `exec()`, ou ouvrir un pipe dans un script setuid ou setgid si `$ENV{PATH}`, `$ENV{IFS}`, `$ENV{CDPATH}`, `$ENV{ENV}` ou `$ENV{BASH_ENV}` sont dérivés de données soumises (ou potentiellement soumises) par l'utilisateur. Le script doit positionner le path à une valeur connue, en utilisant une valeur 'saine'. Voir *perlsec*.

Integer overflow in hex number

(S) Le nombre hexadécimal littéral que vous avez spécifié est trop grand pour votre architecture. Sur une architecture 32-bits le plus grand hexadécimal littéral est 0xFFFFFFFF.

Integer overflow in octal number

(S) Le nombre octal littéral que vous avez spécifié est trop grand pour votre architecture. Sur une architecture 32-bits le plus grand octal littéral est 03777777777.

Internal inconsistency in tracking vforks

(S) Une alerte spécifique à VMS. Perl garde une trace du nombre d'appel à `fork` et `exec`, pour déterminer si l'appel courant à `exec` peut affecter le script en cours ou un sous-process (Voir `exec` in *perlvms*). De façon ou d'autre, ce compte a été bousculé, et Perl fait une conjoncture et traite ce `exec` comme une requête pour terminer le script Perl et exécuter la commande spécifiée.

internal disaster in regexp

(P) Quelque chose s'est très mal passé dans le parseur d'expressions rationnelles.

internal error: glob failed

(P) Il s'est passé quelque chose d'anormal avec le programme externe utilisé avec `glob` et `<*.c>`. Cela veut dire que votre csh (C shell) est rompu. Si c'est cela, vous devez changer toutes les variables relatives à csh dans `config.sh` : si vous avez `tcsh`, faites pointer les variables vers celui-ci comme si c'était csh (ex : `full_csh=/usr/bin/tcsh`); autrement, mettez-les toutes à blanc. (excepté que `d_csh` devrait être 'undef') pour que Perl pense que csh est manquant. Dans ce cas, après avoir édité `config.sh` lancez `./Configure -S` et reconstruisez Perl.

internal urp in regexp at /%/s/

(P) Quelque chose s'est très mal passé dans le parseur d'expressions rationnelles.

invalid [] range in regexp

(F) L'écart spécifié dans la classe de caractères a un caractère minimum supérieur au caractère maximum. Voir *perlre*.

Invalid conversion in %s: "%s"

(W) Perl ne comprend pas le format de conversion donné. Voir `sprintf()` dans *perlfunc*.

Invalid type in pack: '%s'

(F) Le caractère donné n'est pas un type de « pack » valide. Voir `pack()` dans *perlfunc*. (W) Le caractère donné n'est pas un type de « pack » valide mais est utilisé en étant silencieusement ignoré.

Invalid type in unpack: '%s'

(F) Le caractère donné n'est pas un type de « unpack » valide. Voir `unpack()` dans *perlfunc*. (W) Le caractère donné n'est pas un type de « unpack » valide mais est utilisé en étant silencieusement ignoré.

ioctl is not implemented

(F) Votre machine apparemment n'implémente pas `ioctl()`, ce qui est vraiment étonnant pour une machine qui supporte le C.

junk on end of regexp

(P) Le parseur d'expression rationnelle a été embrouillé.

Label not found for "last %s"

(F) Vous nommez une boucle pour en sortir, mais vous n'êtes pas actuellement dans une boucle de ce nom, même si vous comptez d'où vous l'appellez. Voir `last()` dans *perlfunc*.

Label not found for "next %s"

(F) Vous nommez une boucle pour continuer, mais vous n'êtes pas actuellement dans une boucle de ce nom, même si vous comptez d'où vous l'appellez. Voir `next()` dans *perlfunc*.

Label not found for "redo %s"

(F) Vous nommez une boucle pour recommencer, mais vous n'êtes pas actuellement dans une boucle de ce nom, même si vous comptez d'où vous l'appellez. Voir `redo()` dans *perlfunc*.

listen() on closed fd

(W) Vous essayez de faire un `listen()` sur une socket fermée. Peut-être avez-vous oublié de vérifier la valeur retournée par l'appel de `socket()`? Voir `listen()` dans *perlfunc*.

Method for operation %s not found in package %s during blessing

(F) Une tentative a été effectuée pour spécifier une entrée dans une table surchargée qui ne peut résoudre une sous-fonction valide. Voir *overload*.

Might be a runaway multi-line %s string starting on line %d

(S) Un avertissement indiquant que l'erreur précédente peut avoir été causée par un délimiteur manquant dans une chaîne ou un motif (pattern), qui se fermerait éventuellement plus tôt dans la ligne courante.

Misplaced _ in number

(W) Un underscore dans une constante décimale n'est pas sur une limite de 3 chiffres.

Missing \$ on loop variable

(F) Apparemment vous avez programmé en **cs** trop longtemps. Les variables en Perl sont toujours précédées du \$, au contraire des shells, ou cela peut varier d'une ligne à l'autre.

Missing comma after first argument to %s function

(F) Alors que certaines fonctions vous autorisent à spécifier un descripteur de fichier ou un « objet indirect » avant une liste d'argument, celle-ci n'en fait pas partie.

Missing operator before %s?

(S) C'est un message donné à titre indicatif, en conjonction avec le message « %s found where operator expected ». Souvent l'opérateur manquant est le point-virgule.

Missing right bracket

(F) L'analyseur syntaxique a compté plus d'accolades ouvertes que fermées. En règle générale, vous la trouverez à coté de votre dernière modification.

Modification of a read-only value attempted

(F) Vous essayez, directement ou indirectement, de changer la valeur d'une constante. Vous ne pouvez, bien sûr, essayer de faire « $2 = 1$ », car le compilateur l'intercepte. Mais il y a une autre façon de faire la même chose :

```
sub mod { $_[0] = 1 }
mod(2);
```

Un autre moyen serait d'affecter à `substr()` ce qui termine la chaîne.

Modification of non-creatable array value attempted, subscript %d

(F) Vous avez essayé de transformer un élément de tableau en une valeur existante, et l'indice inférieur était probablement négatif, même en comptant à partir de la fin du tableau.

Modification of non-creatable hash value attempted, subscript "%s"

(P) Vous avez essayé de transformer un élément de tableau de hachage en une valeur existante, et il ne peut être créé pour une raison particulière.

Module name must be constant

(F) Seul un nom de module est permis comme premier argument d'un "use".

msg%s not implemented

(F) Vous n'avez pas de messages IPC System V sur votre système.

Multidimensional syntax %s not supported

(W) Les tableaux multidimensionnels ne s'écrivent pas `$foo[1,2,3]`. Ils s'écrivent `$foo[1][2][3]`, comme en C.

Name "%s::%s" used only once: possible typo

(W) Erreur typographique souvent vue si une variable n'est initialisée ou utilisée qu'une seule fois. Si vous avez une bonne raison pour faire cela, mentionnez la variable à nouveau pour supprimer ce message. La ligne `use vars` est fournie pour ce contexte.

Negative length

(F) Vous essayez de faire une opération `read/write/send/recv` avec un buffer de longueur plus petite que 0. C'est difficile à imaginer.

nested *?+ in regexp

(F) Vous ne pouvez quantifier un quantifiant sans faire intervenir de parenthèses. Donc les choses comme `**` ou `+` ou `?*` sont illégales. Remarquez, cependant, que les opérateurs de comparaison `minimum`, `*?`, `++`, et `??` apparaissent comme des quantifiants `nested`, mais ne le sont pas. Voir *perlre*.

No #! line

(F) L'émulateur `setuid` nécessite que le script possède une ligne formée telle que `#!` même si la machine ne reconnaît pas le constructeur `#!`.

No %s allowed while running setuid

(F) Certaines opérations sont supposées être dangereuses pour la sécurité pour un script `setuid` ou `setgid` et ne sont pas autorisées à être lancées. Il y a sûrement un autre moyen pour faire ce que vous voulez, si ce moyen n'est pas sécurisé, il est au moins sécurisable. Voir *perlsec*.

No -e allowed in setuid scripts

(F) Un script `setuid` ne peut être spécifié par l'utilisateur.

No comma allowed after %s

(F) Une liste d'opérateurs qui contient un descripteur de fichier ou un « objet indirect » n'est pas autorisé à avoir une virgule entre lui et les arguments suivants. Autrement, il s'agit juste d'un argument comme un autre.

Une cause possible pour cela est que vous supposez avoir importé une constante dans votre namespace avec `use` ou `import` alors qu'il n'existe pas d'`import` correspondant, par exemple votre système d'exploitation ne supporte pas une constante particulière. En espérant que vous avez bien utilisé une liste explicite des constantes importées que vous vous attendez à voir, voir `use()` dans *perlfunc* et `import()` dans *perlfunc*. Comme une liste explicite d'`import` aurait probablement intercepté cette erreur plus tôt, cela ne remédie pas au fait que votre système d'exploitation ne supporte pas cette constante. Peut-être avez-vous une erreur de frappe dans vos constantes des listes de symboles d'`import` de `use` ou `import` ou dans le nom de constante à la ligne où l'erreur est apparue ?

No command into which to pipe on command line

(F) Une erreur spécifique à VMS. Perl handles its own command line??? redirection, and found a '|' at the end of the command line, so it doesn't know where you want to pipe the output from this command.

No DB::DB routine defined

(F) Le code qui est en train d'être exécuté avec l'option `-d`, mais pour une raison quelconque le fichier `perl5db.pl` (ou un similaire) ne définit pas une routine à appeler en début de chaque déclaration. Ce qui est étrange, car le fichier a été chargé automatiquement, et ne peut passer le `require` s'il ne l'a pas parsé correctement.

No dbm on this machine

(P) C'est compté comme une erreur interne, car chaque machine doit fournir `dbm` de nos jours, car Perl est fourni avec `SDBM`. Voir `SDBM_File`.

No DBsub routine

(F) Le code est exécuté avec l'option **-d**, mais pour une raison quelconque le fichier perl5db.pl (ou un similaire) ne définit pas une routine DB::sub à appeler en début de chaque appel de sous-routines ordinaires.

No error file after 2> or 2>> on command line

(F) Une erreur spécifique à VMS. Perl fait ses propres redirection de ligne de commandes, et a trouvé sur l'entrée standard '2>' ou '2>>', mais ne trouve pas le nom du fichier dans lequel écrire les données destinée à stderr.

No input file after < on command line

(F) Une erreur spécifique à VMS. Perl fait ses propres redirections de ligne de commandes, et a trouvé sur l'entrée standard '<', mais ne trouve pas le nom du fichier dans lequel écrire les données destinées à stdin.

No output file after > on command line

(F) Une erreur spécifique à VMS. Perl fait ses propres redirections de ligne de commandes, et a trouvé sur l'entrée standard '>' en fin de ligne, donc il ne trouve pas où vous voulez rediriger votre sortie.

No output file after > or >> on command line

(F) Une erreur spécifique à VMS. Perl fait ses propres redirections de ligne de commandes, et a trouvé sur l'entrée standard a '>' ou '>>', mais ne trouve pas le nom du fichier dans lequel écrire les données destinées à stdout.

No Perl script found in input

(F) Vous appelez `perl -x`, mais aucune ligne n'est trouvée dans le fichier avec `#!` et contenant le mot « perl ».

No setregid available

(F) Configure n'a rien trouvé qui ressemble à l'appel de `setregid()` pour votre système.

No setreuid available

(F) Configure n'a rien trouvé qui ressemble à l'appel de `setreuid()` pour votre système.

No space allowed after -I

(F) L'argument de **-I** doit suivre **-I** immédiatement après sans qu'aucun espace n'intervienne.

No such array field

(F) Vous essayez d'accéder à un tableau comme avec un tableau associatif, mais le nom de champ utilisé n'est pas défini. Le hash à l'index 0 doit contenir tous les noms de champs valides pour un tableau d'indices pour que cela fonctionne.

No such field "%s" in variable %s of type %s

(F) Vous essayez d'accéder à un champ d'une variable typée où le type ignore le nom du champ. Le nom du champ est recherché dans le tableau associatif `%FIELDS` dans le paquetage `type` à la compilation. Le tableau `%FIELDS` est généralement utilisé avec le pragma `'fields'`.

No such pipe open

(P) Une erreur spécifique à VMS. La routine interne `my_pclose()` a essayé de fermer un pipe qui n'a pas été ouvert. Ceci peut être capturé plus tôt comme une tentative de fermer un descripteur non ouvert.

No such signal: SIG%s

(W) Vous spécifiez un nom de signal comme une sous-fonction d'un sous-script à `%SIG` qu'il ne reconnaît pas. Faire `kill -l` dans votre shell pour voir les noms de signaux valides sur votre système.

Not a CODE reference

(F) Perl essaye d'évaluer une référence vers une valeur de code (c'est cela, une sous-fonction), mais trouve une référence vers quelque chose d'autre à la place. Vous pouvez utiliser la fonction `ref()` pour trouver de quelle référence il s'agit exactement. Voir *perlref*.

Not a format reference

(F) Je ne suis pas sûr de la manière dont vous dirigez la génération d'une référence vers un format anonyme, mais cela indique que vous l'avez fait, et cela n'existe pas.

Not a GLOB reference

(F) Perl essaye d'évaluer une référence vers un « typeglob » (c'est cela, un symbole d'entrée de table qui ressemble à `*foo`), mais trouve une référence vers quelque chose d'autre à la place. Vous pouvez utiliser la fonction `ref()` pour trouver de quelle référence il s'agit exactement. Voir *perlref*.

Not a HASH reference

(F) Perl essaye d'évaluer une référence vers une valeur d'un tableau associatif mais trouve une référence vers quelque chose d'autre à la place. Vous pouvez utiliser la fonction `ref()` pour trouver de quelle référence il s'agit exactement. Voir *perlref*.

Not a perl script

(F) L'émulateur `setuid` requiert que le script ait une ligne `#!` même sur les machines qui ne supportent pas le constructeur `#!`. La ligne doit au moins mentionner `perl`.

Not a SCALAR reference

(F) Perl essaye d'évaluer une référence vers un scalaire, mais trouve une référence vers quelque chose d'autre. Vous pouvez utiliser la fonction `ref()` pour trouver de quelle référence il s'agit exactement. Voir *perlref*.

Not a subroutine reference

(F) Perl essaye d'évaluer une référence vers une valeur de code (c'est cela, une sous-fonction), mais trouve une référence vers quelque chose d'autre à la place. Vous pouvez utiliser la fonction `ref()` pour trouver de quelle référence il s'agit exactement. Voir *perlref*.

Not a subroutine reference in overload table

(F) Une tentative a été faite pour spécifier une entrée dans une table surchargée qui ne pointe pas vers une sous-fonction valide. Voir *overload*.

Not an ARRAY reference

(F) Perl essaye d'évaluer une référence vers une valeur de tableau, mais trouve une référence vers quelque chose de différent. Vous pouvez utiliser la fonction `ref()` pour trouver de quel type de référence il s'agit vraiment. Voir *perlref*.

Not enough arguments for %s

(F) La fonction nécessite plus d'arguments que vous n'avez spécifié.

Not enough format arguments

(W) Le format spécifie plus de champs d'images que la ligne suivante ne lui en fournit. Voir *perform*.

Null filename used

(F) Vous ne pouvez charger un fichier avec un nom nul, car sur certaines machines cela veut dire le répertoire courant ! Voir `require()` dans *perlfunc*.

Null picture in formline

(F) Le premier argument de `formline` doit être un format de spécification d'image valide. Cet argument est vide, ce qui veut probablement dire que vous lui avez soumis une valeur non initialisée. Voir *perform*.

NULL OP IN RUN

(P) Une fonction interne a appelé `run()` avec un pointer null opcode.

Null realloc

(P) Une tentative a eu lieu pour réallouer NULL.

NULL regexp argument

(P) La fonction interne de recherche de motif a été soufflé.

NULL regexp parameter

(P) La fonction interne de recherche de motif a été soufflée. (Ndt : Hors de sa courge dans la version originale)

Number too long

(F) Perl limite la représentation des nombres décimaux dans le programme à 250 caractères. Vous avez dépassé la limite. Les versions futures de Perl vont éliminer cette limitation arbitraire. En attendant, essayez d'utiliser la notation scientifique (ex. « 1e6 » au lieu de « 1_000_000 »).

Odd number of elements in hash assignment

(S) Vous avez spécifié un nombre impair d'éléments pour initialiser votre tableau indexé, ce qui est bizarre car un tableau indexé est utilisé avec des paires clé/valeur.

Offset outside string

(F) Vous essayez de faire une opération `read/write/send/rcv` operation avec un offset qui pointe en dehors du buffer. Cela est difficile à imaginer. La seule exception à cela est que `sysread()` past the buffer will extend the buffer and zero pad the new area.???

oops: oopsAV

(S) Un avertissement interne que la grammaire est vissée vers le haut. ???

oops: oopsHV

(S) Un avertissement interne que la grammaire est vissée vers le haut.

Operation '%s': no method found, %s

(F) une tentative a été faite pour surcharge d'opérations pour laquelle il n'existe pas de descripteur de fichier défini. Alors que certains descripteurs peuvent être automatiquement régénérés dans les conditions d'autres descripteurs, il n'y pas de descripteur par défaut pour quelque opération que ce soit, à moins que la clé `fallback` ne soit spécifiée à vrai. Voir *overload*.

Operator or semicolon missing before %s

(S) Vous utilisez une variable ou un appel de fonction là où le parseur s'attend à trouver un opérateur. Le parseur suppose que vous pensiez vraiment utiliser un opérateur, mais cela est fortement vu comme incorrect. Par exemple, si vous dites « `*foo *foo` » cela est interprété comme si vous disiez « `*foo * 'foo'` ».

Out of memory for yacc stack

(F) Le parseur yacc recherche à agrandir sa pile donc il peut continuer, mais `realloc()` ne veut pas lui donner plus de mémoire, virtuelle ou autre.

Out of memory during request for %s

(X|F)(F) La fonction `malloc()` a retourné 0, ce qui indique un manque de mémoire (ou de mémoire virtuelle) pour satisfaire la requête. La requête est jugée petite, ce qui rend la possibilité de capturer l'erreur dépendante de la façon dont Perl a été compilé. Par défaut, cela ne peut être capturé. Cependant, si compilé à cet effet, Perl peut utiliser le contenu de `$_M` comme une sortie d'urgence après le `die()` et son message. Dans ce cas l'erreur peut-être capturée.

Out of memory during "large" request for %s

(F) La fonction `malloc()` a retourné 0, ce qui indique un manque de mémoire (ou de mémoire virtuelle) pour satisfaire la requête. Cependant, la requête a été jugée assez large (par défaut 64K), donc la possibilité de capturer cette erreur est permise.

Out of memory during ridiculously large request

(F) Vous ne pouvez allouer plus de $2^{31}+1$ octets. Cette erreur est généralement causée par une faute de frappe dans le programme Perl. Ex, `$arr[time]` au lieu de `$arr[$time]`.

page overflow

(W) Un appel simple de `write()` a produit plus de lignes que peut contenir une page. Voir *perlforn*.

panic: ck_grep

(P) Echec d'un test de cohérence lors de la compilation d'un `grep`.

panic: ck_split

(P) Echec d'un test de cohérence lors de la compilation d'un `split`.

panic: corrupt saved stack index

(P) La sauvegarde de la pile a été appelée pour restaurer plus de valeurs qu'il n'y en a dans la pile.

panic: die %s

(P) On est passé du contexte de pile à un contexte de `eval`, pour finalement découvrir qu'on n'est pas dans un contexte d'`eval`.

panic: do_match

(P) La fonction interne `pp_match()` a été appelé avec des données opérationnelles non valides.

panic: do_split

(P) Quelque chose de grave est arrivé lors de la préparation du `split`.

panic: do_subst

(P) La fonction interne `pp_subst()` a été appelé avec des données opérationnelles non valides.

panic: do_trans

(P) La fonction interne `do_trans()` a été appelé avec des données opérationnelles non valides.

panic: frexp

(P) La fonction `frexp()` de la `library` a échoué, rendant le `printf("%f")` impossible.

panic: goto

(P) On a déchargé le contexte de pile vers le contexte avec l'étiquette (`label`) spécifiée, et ensuite découvert que c'était une étiquette que l'on ne sait pas joindre.

panic: INTERPCASEMOD

(P) L'analyseur syntaxique est arrivé dans ce mauvais état au moment d'un `case` modifier.???

panic: INTERPCONCAT

(P) L'analyseur syntaxique est dans un mauvais état après avoir parsé une chaîne de caractères avec des parenthèses.

panic: last

(P) On a déchargé le contexte de pile vers un contexte de bloc, et découvert par la suite que ce n'est pas un contexte de bloc.

panic: leave_scope clears

(P) Une variable lexicale est maintenant en lecture seule d'une manière ou d'une autre dans le bloc.

panic: leave_scope inconsistency

(P) La sauvegarde de pile n'est sûrement plus synchronisée. À la fin, il y a un `type enum` invalide en haut du tout.

panic: malloc

(P) Quelque chose a demandé un nombre négatif d'octets dans `malloc`.

panic: mapstart

(P) Le compilateur s'est embrouillé avec la fonction map().

panic: null array

(P) L'un des tableaux internes de fonctions a passé un pointeur AV null

panic: pad_alloc

(P) Le compilateur s'est embrouillé au sujet de quelle zone de travail où affecter les valeurs temporaires et lexicales.

panic: pad_free curpad

(P) Le compilateur s'est embrouillé au sujet de quelle zone de travail où affecter les valeurs temporaires et lexicales.

panic: pad_free po

(P) Un décalage incorrect de zone de travail a été détecté en interne.

panic: pad_reset curpad

(P) Le compilateur s'est embrouillé au sujet de quelle zone de travail où affecter les valeurs temporaires et lexicales.

panic: pad_sv po

(P) Un décalage incorrect de zone de travail a été détecté en interne.

panic: pad_swipe curpad

(P) Le compilateur s'est embrouillé au sujet de quelle zone de travail où affecter les valeurs temporaires et lexicales.

panic: pad_swipe po

(P) Un décalage incorrect de zone de travail a été détecté en interne.

panic: pp_iter

(P) L'itérateur foreach n'a pas été appelé dans un contexte de boucle.

panic: realloc

(P) Quelque chose a demandé un nombre négatif d'octets à realloc.

panic: restartop

(P) Une routine interne a demandé un goto (ou quelque chose de semblable), et n'a pas fourni de destination.

panic: return

(P) On est passé du contexte de pile à un contexte de sous-fonction ou eval, pour finalement découvrir qu'on n'est pas dans un contexte d'eval ou de sous-fonction.

panic: scan_num

(P) scan_num() a été appelé avec quelque chose qui n'est pas un nombre.

panic: sv_insert

(P) La routine sv_insert() a été utilisé pour enlever plus de chaînes qu'il n'en existe.

panic: top_env

(P) Le compilateur s'attendait à faire un goto, ou quelque chose d'équivalent.

panic: yylex

(P) L'analyseur syntaxique est dans un état invalide pendant qu'il était en train de procéder à un case.

Parentheses missing around "%s" list

(W) Vous dites quelque chose comme

```
my $foo, $bar = @_;
```

alors que vous pensez

```
my ($foo, $bar) = @_;
```

Souvenez-vous que « my » et « local » doivent être délimités par des parenthèses.

Perl %3.3f required--this is only version %s, stopped

(F) Le module en question utilise une possibilité offerte par une version plus récente que celle exécutée actuellement. Depuis combien de temps votre Perl n'a-t-il pas été mis à jour ? Voir require() dans *perlfunc*.

Permission denied

(F) L'émulateur suidperl voit qui vous êtes et interdit l'exécution.

pid %d not a child

(W) Une alerte spécifique à VMS. Waitpid() a été appelée pour attendre un processus qui n'est pas un sous-processus du processus courant. Même si cela est bien sous la perspective VMS, cela n'est probablement pas ce que vous voulez.

POSIX getpgrp can't take an argument

(F) Votre compilateur C utilise la fonction POSIX `getpgrp()`, qui ne prend pas d'argument, au contraire de la version BSD, qui prend un pid.

Possible attempt to put comments in `qw()` list

(W) Une liste par `qw()` ne contient que des éléments séparés par des espaces; comme avec les chaînes littérales, les caractères de commentaires ne sont pas ignorés, mais au lieu de ça traités comme des données littérales. (Vous pouvez utiliser un délimiteur différent que les parenthèses indiquées ici. Les crochets sont fréquemment utilisés.)

Vous avez probablement écrit quelque chose comme cela :

```
@list = qw(
    a # a comment
    b # another comment
);
```

alors qu'il devrait être écrit comme cela :

```
@list = qw(
    a
    b
);
```

Si vous voulez vraiment des commentaires, construisez votre liste à l'ancienne façon, avec des cotes et des virgules :

```
@list = (
    'a',    # un commentaire
    'b',    # un autre commentaire
);
```

Possible attempt to separate words with commas

(W) Les listes `qw()` contiennent des éléments séparés par des espaces. Donc, les virgules ne sont pas nécessaires pour séparer les éléments. (Vous pouvez utiliser un délimiteur différent des parenthèses montrées ici ; les crochets sont fréquemment utilisés.)

Vous avez écrit probablement quelque chose comme ça :

```
qw! a, b, c !;
```

Ce qui place les caractères virgules comme éléments de la liste. Écrivez ceci sans virgule si vous ne voulez pas que cela apparaisse dans vos données :

```
qw! a b c !;
```

Possible memory corruption: `%s` overflowed 3rd argument

(F) Un appel à `ioctl()` ou `fcntl()` a retourné plus que ce Perl peut traiter. Perl consacre un tampon raisonnable, mais place un octet sentinelle à la fin du tampon au cas où. Le bit sentinelle a été dépassé, et Perl suppose que sa mémoire est maintenant corrompue. Voir `ioctl()` dans *perlfunc*.

Precedence problem: `open %s` should be `open(%s)`

(S) L'ancienne construction irrégulière

```
open FOO || die;
```

est maintenant mal interprétée comme

```
open(FOO || die);
```

à cause de la stricte régularisation de la grammaire de Perl 5 en opérateurs unaires et listes. (L'ancien était un peu des deux.) Vous devez mettre une parenthèse autour du descripteur de fichier, ou utiliser le nouvel opérateur « `or` » à la place de « `||` ».

print on closed filehandle `%s`

(W) Le descripteur de fichier dans lequel vous essayez d'imprimer a été fermé quelque temps auparavant. Vérifiez votre flux de données.

printf on closed filehandle `%s`

(W) Le descripteur de fichier dans lequel vous essayez d'écrire a été fermé quelque temps auparavant. Vérifiez votre flux de données.

Probable precedence problem on %s

(W) Le compilateur a trouvé un simple mot là où il attendait un opérateur conditionnel, ce qui indique souvent que le `||` ou `&&` a été parsé comme une partie du dernier argument du constructeur, par exemple :

```
open FOO || die;
```

Prototype mismatch: %s vs %s

(S) La sous-fonction qui est en train d'être déclarée ou définie a été précédemment déclarée ou définie avec un prototype différent.

Range iterator outside integer range

(F) Un (ou deux) arguments numériques de l'opérateur d'échelles « .. » est en dehors des bornes qui peuvent être définies par des entiers en interne. Un moyen de détourner cela est de forcer Perl à utiliser les chaînes de caractères « magicals » incrémentées en ajoutant « 0 » à vos nombres.

Read on closed filehandle <%s>

(W) Le descripteur de fichier a été fermé avant cette opération. Vérifiez votre flux logique de données.

Reallocation too large: %lx

(F) Vous ne pouvez allouer plus de 64K sur une machine MS-DOS.

Recompile perl with -DDEBUGGING to use -D switch

(F) Vous ne pouvez utiliser l'option `-D` à moins que le code pour produire la sortie désirée ait été compilé dans Perl, ce qui implique certaines contraintes non disponibles dans votre version de Perl.

Recursive inheritance detected in package '%s'

(F) Au moins 100 niveaux d'héritage ont été utilisés. Cela indique probablement une boucle inattendue dans votre hiérarchie de classes.

Recursive inheritance detected while looking for method '%s' in package '%s'

(F) Au moins 100 niveaux d'héritage ont été rencontrés lors de l'appel de méthodes. Cela indique probablement une boucle inattendue dans votre hiérarchie de classes.

Reference found where even-sized list expected

(W) Vous avez donné une référence unique là où Perl s'attend à avoir une liste avec un même nombre d'éléments (pour affectation à un tableau associatif). Cela veut dire que vous utilisez le constructeur anon hash alors que vous vous attendiez à utiliser parens. Dans tous les cas, un tableau associatif (hash) nécessite une **paire** clef/valeur.

```
%hash = { one => 1, two => 2, }; # MAUVAIS
%hash = [ qw/ an anon array / ]; # MAUVAIS
%hash = ( one => 1, two => 2, ); # correct
%hash = qw( one 1 two 2 ); # correct
```

Reference miscount in sv_replace()

(W) La fonction interne `sv_replace()` a levé une nouvelle SV avec un nombre de références différent de 1.

regexp *+ operand could be empty

(F) La partie d'un motif (regexp) sujet à l'opérateur de quantification `*` ou `+` doit être une chaîne vide.

regexp memory corruption

(P) Le moteur d'expression rationnelle a été mis en confusion par ce que l'expression rationnelle lui a donnée.

regexp out of space

(P) Une erreur « qui ne peut pas arriver », car `safemalloc()` aurait dû la capturer plus tôt.

regexp too big

(F) L'implémentation courante des expressions rationnelles utilise des shorts comme offset d'adresse dans une chaîne. Malheureusement cela veut dire que si l'expression rationnelle compilée est plus longue que 32767, il s'arrête. Généralement quand vous voulez une expression rationnelle aussi longue, il est mieux de passer par des déclarations multiples. Voir *perlre*.

Reversed %s= operator

(W) Vous écrivez votre opérateur d'affectation à l'envers. Le `=` doit toujours arriver en dernier, pour éviter l'ambiguïté avec les opérateurs unaires suivants.

Runaway format

(F) Votre format contient la séquence `~~` repeter-jusqu'à-la-sequence-de-blanc, mais cela produit 200 lignes au moins, et la 200^e ligne apparaît exactement comme la 199^e. Apparemment vous n'avez pas arrangé les arguments pour qu'ils s'épuisent, soit en utilisant `^` au lieu de `@` (pour les variables scalaires), soit en faisant un `shift` ou un `pop` (pour les tableaux). Voir *perlform*.

Scalar value @%s[%s] better written as \$%s[%s]

(W) Vous avez utilisé une tranche de tableau (indiqué par @) pour sélectionner un élément unique d'un tableau. Généralement, il est mieux de demander pour une valeur scalaire (indiquée par \$). La différence est que \$foo[&bar] se comporte toujours comme un scalaire, à la fois au moment de l'affectation et quand vous l'évaluez, alors que @foo[&bar] se comporte comme une liste lorsque vous l'affectez, et fournit une liste dans un sous-script, ce qui peut faire des choses étranges si vous vous attendiez à seulement un sous-script. En d'autres termes, si vous espérez actuellement traiter l'élément de tableau associatif comme une liste, vous devez regarder dedans comment les références fonctionnent, car Perl ne va pas faire la conversion entre les scalaires et les listes pour vous. Voir *perlref*.

Scalar value @%s{%s} better written as \$%s{%s}

(W) Vous avez utilisé une tranche de tableau (indiqué par @) pour sélectionner un élément unique d'un tableau. Généralement, il est mieux de demander pour une valeur scalaire (indiquée par \$). La différence est que \$foo{&bar} se comporte toujours comme un scalaire, à la fois au moment de l'affectation et quand vous l'évaluez, alors que @foo{&bar} se comporte comme une liste lorsque vous l'affectez, et fourni une liste dans un sous-script, ce qui peut faire des choses étranges si vous vous attendiez à seulement un sous-script.

En d'autres termes, si vous espérez actuellement traiter l'élément de tableau associatif comme une liste, vous devez regarder dedans comment les références fonctionnent, car Perl ne va pas faire la conversion entre les scalaires et les listes pour vous. Voir *perlref*.

Script is not setuid/setgid in suidperl

(F) Bizarrement, le programme suidperl a été invoqué dans un script qui n'a pas de bit setuid ou setgid bit positionné. Cela n'a vraiment pas de sens.

Search pattern not terminated

(F) L'analyseur syntaxique ne peut trouver le délimiteur final du constructeur de // ou m{ }. Souvenez-vous que les parenthèses comptent les niveaux voisins. Manquer un \$ d'une variable \$m peut causer cette erreur.

%sseek() on unopened file

(W) Vous essayez d'utiliser un seek() ou sysseek() sur un descripteur de fichier qui n'a jamais été ouvert ou qui a été fermé depuis.

select not implemented

(F) Cette machine n'implémente pas l'appel système select().

sem%s not implemented

(F) Vous n'avez pas de sémaphore IPC System V sur votre système.

semi-panic: attempt to dup freed string

(S) La routine interne newSVsv() a été appelée pour dupliquer un scalaire qui a été marqué comme libre depuis.

Semicolon seems to be missing

(W) Une erreur de syntaxe a été probablement causée par un point-virgule manquant ou un autre opérateur manquant, comme une parenthèse.

Send on closed socket

(W) Le descripteur de fichier a été fermé avant cette opération. Vérifiez votre flux logique de données.

Sequence (? incomplete

(F) Une expression rationnelle est terminée par une extension incomplète. (?). Voir *perlre*.

Sequence (?#... not terminated

(F) Une expression rationnelle doit se terminer par une parenthèse fermée. Les parenthèses incluses ne sont pas autorisées. Voir *perlre*.

Sequence (?%s...) not implemented

(F) L'extension de l'expression rationnelle proposée a le mot réservé mais n'a pas encore été écrite. Voir *perlre*.

Sequence (?%s...) not recognized

(F) Vous utilisez une expression rationnelle qui n'a pas de sens. Voir *perlre*.

Server error (500 Server error)

Connu sous le nom de « 500 Server error ». **Ceci est une erreur CGI, pas une erreur Perl.** Vous devez être sûr que votre script est exécutable, accessible par l'utilisateur CGI qui lance le script (qui n'est probablement pas l'utilisateur que vous utilisez pour faire vos tests), que vous n'utilisez pas de variables d'environnement (comme PATH) que l'utilisateur CGI ne va pas avoir, et que vous n'utilisez pas de ressources injoignables pour le serveur. Voir pour plus d'information :

<http://www.perl.com/perl/faq/idiots-guide.html>
<http://www.perl.com/perl/faq/perl-cgi-faq.html>
<ftp://rtfm.mit.edu/pub/usenet/news.answers/www/cgi-faq>
<http://hoohoo.ncsa.uiuc.edu/cgi/interface.html>
<http://www-genome.wi.mit.edu/WWW/faqs/www-security-faq.html>

setegid() not implemented

(F) Vous essayez d'affecter \$), et votre système d'exploitation ne supporte pas l'appel système setegid() (ou un équivalent), ou du moins c'est ce que Configure pense.

seteuid() not implemented

(F) Vous essayez de référencer \$>, et votre système d'exploitation ne supporte pas l'appel système seteuid() (ou un équivalent), ou au moins Configure n'y a pas pensé.

setrgid() not implemented

(F) Vous essayez de référencer \$(), et votre système d'exploitation ne supporte pas l'appel système setrgid() (ou un équivalent), ou au moins Configure n'y a pas pensé.

setruid() not implemented

(F) Vous essayez de référencer \$<, et votre système d'exploitation ne supporte pas l'appel système setruid() (ou un équivalent), ou au moins Configure n'y a pas pensé.

Setuid/gid script is writable by world

(F) L'émulateur setuid ne va pas lancer un script qui est modifiable par le monde, car le monde peut très bien déjà l'avoir modifié.

shm%s not implemented

(F) Vous n'avez pas de System V shared memory IPC sur votre système.

shutdown() on closed fd

(W) Vous essayez de faire un shutdown sur socket fermée. Cela paraît un peu superflu.

SIG%s handler "%s" not defined

(W) Le gestionnaire de signal nommé dans %SIG n'existe pas. Peut-être l'avez-vous défini dans le mauvais paquetage ?

sort is now a reserved word

(F) Un message d'erreur ancien que personne ne lancera plus. Car avant que 'sort' soit un mot-clef, on l'utilisait parfois comme un descripteur de fichier.

Sort subroutine didn't return a numeric value

(F) Une comparaison avec la fonction sort() doit retourner un nombre. Cela est arrivé en utilisant <=> ou cmp, ou en ne les utilisant pas correctement. Voir sort() dans *perlfunc*.

Sort subroutine didn't return single value

(F) Une comparaison faite avec le sous-programme sort() peut ne pas retourner une liste avec plus ou moins d'un élément. Voir sort() dans *perlfunc*.

Split loop

(P) Un split boucle indéfiniment. (Évidemment, un split ne peut boucler plus de fois qu'il y a de caractères en entrée, ce qui est arrivé.) Voir split() dans *perlfunc*.

Stat on unopened file <%s>

(W) Vous essayez d'utiliser la fonction stat() (ou une fonction de test de fichier équivalent) sur un descripteur de fichier qui n'a jamais été ouvert ou qui a été fermé depuis.

Statement unlikely to be reached

(W) Vous faites un exec() avec certaines déclarations après autre qu'un die()???. Cela est toujours une erreur car exec() ne retourne jamais rien à moins d'une erreur. Vous voulez probablement utiliser system() à la place, qui lui retourne. Pour supprimer ce warning, mettez le exec dans un bloc tout seul.

Stub found while resolving method '%s' overloading '%s' in package '%s'

(P) Surcharger la résolution au dessus de l'arbre @ISA peut être rompue par l'importation de stubs. Les stubs ne peuvent jamais être implicitement créés, mais un appel explicite à can peut rompre cela.

Subroutine %s redefined

(W) Vous redéfinissez un sous-programme. Pour supprimer ce message, faites

```
{
    local $^W = 0;
    eval "sub name { ... }";
}
```

Substitution loop

(P) La substitution boucle indéfiniment. (Évidemment, une substitution ne peut itérer plus de fois qu'il y a de caractères en entrée, ce qui est arrivé.) Voir la discussion sur les substitutions dans Opérateurs apostrophe et type apostrophe in *perlop*.

Substitution pattern not terminated

(F) L'analyseur syntaxique ne peut trouver le délimiteur intérieur du constructeur de *s///* ou *s{ }{ }*. Souvenez-vous que les parenthèses délimitent le nombre de voisins. Un *\$* manquant sur une variable *\$s* peut causer cette erreur.

Substitution replacement not terminated

(F) L'analyseur syntaxique ne peut trouver le délimiteur intérieur du constructeur de *s///* ou *s{ }{ }*. Souvenez-vous que les parenthèses délimitent le nombre de voisins. Un *\$* manquant sur une variable *\$s* peut causer cette erreur.

substr outside of string

(S),(W) Vous essayez de faire référence à la fonction *substr()* qui pointe en dehors d'une chaîne. En fait la valeur absolue de l'offset est plus grand que la longueur de la chaîne. Voir *substr()* dans *perlfunc*. Cette alerte est impérative si *substr* est utilisé dans un contexte de lvalue (comme opérateur du côté gauche de l'affectation, ou comme argument à une sous-fonction par exemple.)

suidperl is no longer needed since %s

(F) Votre Perl a été compilé avec **-DSETUID_SCRIPTS_ARE_SECURE_NOW**, mais une version de l'émulateur *setuid* est tout de même arrivée à se lancer.

syntax error

(F) Cela veut probablement dire que vous avez une erreur de syntaxe. Les raisons les plus probables sont :

- Un mot clef est mal écrit.
- Un point-virgule est manquant.
- Une virgule est manquante.
- Une parenthèse ouverte ou fermée est manquante.
- Un crochet ouvrant ou fermant est manquant.
- Il manque une cote.

Souvent, il y a un autre message associé avec l'erreur de syntaxe qui donne plus d'information. (Des fois cela aide d'activer **-w**.) Le message d'erreur en lui-même dit souvent à quelle ligne il s'est arrêté. Des fois l'erreur actuelle est bien avant, car Perl est fort pour la compression d'entrée au hasard. Occasionnellement le numéro de ligne peut être inexact, et la seule manière de savoir ce qui se passe est d'appeler de façon répétitive *perl -c*, en découpant à la moitié du programme à chaque fois pour voir où l'erreur apparaît. Une sorte de version cybernétique de 20 questions.

syntax error at line %d: '%s' unexpected

(A) Vous lancez accidentellement votre script à travers le Bourne shell à la place de Perl. Vérifiez la ligne *#!*, ou positionnez manuellement votre script dans Perl vous-même.

System V %s is not implemented on this machine

(F) Vous essayez de faire quelque chose avec une fonction commençant par « *sem* », « *shm* », ou « *msg* » mais ce System V IPC n'est pas implémenté sur votre machine. Sur certaines machines la fonctionnalité peut exister mais n'est pas configurée. Consulter votre support système.

Syswrite on closed filehandle

(W) Le descripteur de fichier dans lequel vous écrivez a été fermé quelques????? auparavant. Vérifiez votre flux logique de données.

Target of goto is too deeply nested

(F) Vous essayez d'atteindre une étiquette en utilisant *goto*, une étiquette qui est trop loin pour que Perl puisse l'atteindre. Perl vous fait une faveur en vous le refusant.

tell() on unopened file

(W) Vous essayez d'utiliser la fonction *tell()* sur un descripteur de fichier qui n'a jamais été ouvert ou qui a été fermé depuis.

Test on unopened file <%s>

(W) Vous essayez d'invoquer un opérateur de test de fichier sur un descripteur de fichier qui n'est pas ouvert. Vérifiez votre logique. Voir aussi **-X** in *perlfunc*.

That use of \$[is unsupported

(F) L'affectation de *\$[* est maintenant strictement réglementée, et interprétée comme une directive du compilateur. Vous devez maintenant seulement avoir une solution parmi celles-ci :


```

    $[ = 0;
    $[ = 1;
    ...
    local $[ = 0;
    local $[ = 1;
    ...

```

Cela est pour prévenir le problème d'un module changeant la base du tableau depuis un autre module par inadvertance. Voir \$[in *perlvar*.

The %s function is unimplemented

La fonction indiquée n'est pas implémentée sur cette architecture, en accord avec les choix de Configure.

The crypt() function is unimplemented due to excessive paranoia

(F) Configure ne peut trouver la fonction `crypt()` sur votre machine, probablement parce que votre vendeur ne l'a pas fournie, probablement pas qu'il pense que c'est un secret, ou du moins ils prétendent que ça continue d'être le cas. Et si vous citez mes paroles, je les dénierai.

The stat preceding -1 _ wasn't an lstat

(F) Cela n'a pas de sens de tester le tampon courant de stat pour un lien symbolique si le dernier stat qui a écrit dans le tampon a déjà passer le lien symbolique pour obtenir le fichier réel. Utilisez un autre nom de fichier à la place.

times not implemented

(F) Votre version de librairie C ne fait pas apparemment de `times()`. Je suspecte que vous n'êtes pas sous Unix.

Too few args to syscall

(F) Il doit y avoir au moins un argument à la fonction `syscall()` pour spécifier l'appel système à appeler, étourdi.

Too late for "-T" option

(X) La ligne `#!` (ou l'équivalent `local`) dans un script Perl contient l'option `-T`, mais Perl n'a pas été invoquée avec `-T` en ligne de commande. C'est une erreur car, quand Perl découvre le `-T` dans un script, il est trop tard pour tout teinter???? dans l'environnement. Donc Perl rend la main.

Si le script Perl a été exécuté comme une commande utilisant le `#!` mécanisme (ou son équivalent `local`), cette erreur peut être éventuellement corrigée en éditant la ligne `#!` pour que l'option `-T` soit une part du premier argument de Perl : ex. Changer `perl -n -T en perl -T -n`.

Si le script Perl a été exécuté avec `perl scriptname`, alors l'option `-T` doit apparaître sur la ligne de commande : `perl -T scriptname`.

Too late for "-%s" option

(X) La ligne `#!` (ou l'équivalent `local`) dans un script Perl contient l'option `-M` ou `-m`. C'est une erreur car les options `-M` et `-m` ne sont pas prévues pour être utilisées dans un script. Utilisez `use` à la place.

Too many ('s

Too many)'s

(A) Vous lancez accidentellement votre script à travers `csh` au lieu de Perl. Vérifiez la ligne `#!`, ou soumettez manuellement votre script à Perl vous-même.

Too many args to syscall

(F) Perl supporte un maximum de seulement 14 arguments pour `syscall()`.

Too many arguments for %s

(F) La fonction demande moins d'arguments que ceux que vous avez spécifiés.

trailing \ in regexp

(F) L'expression rationnelle se termine par une bloque oblique verticale toute seule (au lieu de deux). Voir *perlre*.

Transliteration pattern not terminated

(F) L'analyseur syntaxique ne peut trouver le délimiteur intérieur d'une construction de type `tr///` ou `tr[][]` ou `y///` ou `y[][]`. Le `$` manquant devant les variables `$tr` ou `$y` peuvent causer cette erreur.

Transliteration replacement not terminated

(F) L'analyseur syntaxique ne peut trouver le délimiteur final d'une construction de type `tr///` ou `tr[][]`.

truncate not implemented

(F) Votre machine n'implémente pas de mécanisme de troncation de fichier que Configure peut reconnaître.

Type of arg %d to %s must be %s (not %s)

(F) Cette fonction requiert que l'argument de cette position soit d'un certain type. Les tableaux doivent être `@NAME` ou `@{EXPR}`. Les tableaux associatifs doivent être `%NAME` ou `%{EXPR}`. Les effacements de références de façon implicite ne sont pas permis. Utilisez la forme `{EXPR}` comme effacements de références explicites. Voir *perlref*.

umask: argument is missing initial 0

(W) Un umask de 222 est incorrect. Cela peut-être 0222, car les littéraires octaux commencent toujours par 0 en Perl, comme en C.

umask not implemented

(F) Votre machine ne semble pas implémenter la fonction umask et vous essayez de l'utiliser pour restreindre les permissions pour vous-même. (EXPR & 0700).

Unable to create sub named "%s"

(F) Vous essayez de créer ou d'accéder à une fonction avec un nom illégal.

Unbalanced context: %d more PUSHes than POPs

(W) Le code de sortie a détecté un problème interne dans le nombre de contextes dans lequel il est entré et sorti.

Unbalanced saves: %d more saves than restores

(W) Le code de sortie a détecté un problème interne dans le nombre de valeurs qui ont été temporairement localisées.

Unbalanced scopes: %d more ENTERs than LEAVEs

(W) Le code de sortie a détecté un problème interne dans le nombre de blocs dans lequel il est entré et sorti.

Unbalanced tmps: %d more allocs than frees

(W)(W) Le code de sortie a détecté un problème interne dans le nombre de scalaires mortal qui ont été alloués et libérés.

Undefined format "%s" called

(F) Le format indiqué ne semble pas exister peut-être est-ce dans un autre paquetage ? Voir *perlform*.

Undefined sort subroutine "%s" called

(F) La fonction de comparaison de sort spécifiée ne semble pas exister. Peut-être est-ce dans un autre paquetage ? Voir `sort ()` dans *perlfunc*.

Undefined subroutine &%s called

(F) La sous-fonction indiquée que vous essayez d'appeler n'a pas été définie, ou si elle l'a été, elle a été indéfinie depuis.

Undefined subroutine called

(F) La sous-fonction anonyme que vous essayez d'appeler n'a pas été définie, ou si elle l'a été, elle a été indéfinie depuis.

Undefined subroutine in sort

(F) La fonction de comparaison de sort spécifiée est déclarée mais ne semble pas avoir été définie pour le moment. Voir `sort ()` dans *perlfunc*.

Undefined top format "%s" called

(F) Le format indiqué ne semble pas exister. Peut-être est-ce en réalité dans un autre paquetage ? Voir *perlform*.

Undefined value assigned to typeglob

(W) Une valeur indéfinie a été assignée à un typeglob, avec `*foo = undef`. Cela ne veut rien dire. Il est possible que vous pensiez en réalité à `undef *foo`.

unexec of %s into %s failed!

(F) La fonction `unexec()` a échoué pour une quelconque raison. Voir votre représentant FSF, qui vous a probablement mis cela en place le premier.

Unknown BYTEORDER

(F) Il n'y a pas de fonctions de swap de bits avec une machine avec cet ordre d'octets.

unmatched () in regexp

(F) Les parenthèses non précédées de backslash doivent toujours être équilibrées dans les expressions rationnelles. Si vous êtes utilisateur de vi, la touche % est utilisée pour trouver la parenthèse correspondante. Voir *perlre*.

Unmatched right bracket

(F) L'analyseur syntaxique a compté plus d'accolades fermées que d'ouvertes, donc vous avez probablement oublié d'en mettre une près de l'endroit que vous avez édité en dernier.

unmatched [] in regexp

(F) Les crochets autour d'une classe de caractères doivent se correspondre. Si vous voulez inclure un crochet fermant dans une classe de caractères, mettez la barre oblique inverse devant, ou mettez-la en premier. Voir *perlre*.

Unquoted string "%s" may clash with future reserved word

(W) Vous utilisez un simple mot qui peut être utilisé parfois comme mot réservé. Il est mieux de mettre un tel mot entre cotes, ou en lettres capitales, ou d'insérer un underscore dans son nom. Vous pouvez également le déclarer comme une sous-fonction.

Unrecognized character %s

(F) Le parseur Perl n'a aucune idée pour quoi faire avec le caractère spécifié dans votre script Perl (ou eval). Peut-être essayez-vous de lancer un script compressé, ou un programme binaire, ou un répertoire comme un programme Perl.

Unrecognized signal name "%s"

(F) Vous spécifiez un nom de signal à la fonction `kill()` qui n'est pas reconnue. Dites `kill -l` dans votre shell pour voir les signaux valides sur votre système.

Unrecognized switch: -%s (-h will show valid options)

(F) Vous avez spécifié une option illégale pour Perl. Ne faites pas cela. (Si vous ne pensez pas faire cela, vérifiez la ligne `#!` pour voir si vous n'avez pas spécifié la mauvaise option.)

Unsuccessful %s on filename containing newline

(W) Une opération de fichier a été tenté sur un nom de fichier, et cette opération a échoué, **PROBABLEMENT** parce que le nom de fichier contient un caractère de nouvelle ligne, **PROBABLEMENT** parce que vous avez oublié de faire un `chomp()` ou un `chomp()`. Voir `chomp()` dans *perlfunc*.

Unsupported directory function "%s" called

(F) Votre machine ne supporte pas `opendir()` et `readdir()`.

Unsupported function fork

(F) Votre version d'exécutable ne supporte pas `fork()`.

Remarquez que sur certains systèmes, comme OS/2, il peut y avoir différentes versions de l'exécutable Perl, certains supportent `fork` d'autres non. Essayez de changer le nom que vous utilisez pour appeler Perl par `perl_`, à `perl_`, et ainsi de suite.

Unsupported function %s

(F) Votre machine n'implémente pas la fonction indiquée, apparemment. Ou du moins, Configure pense cela.

Unsupported socket function "%s" called

(F) Votre machine ne supporte pas le mécanisme des sockets Berkeley, Ou du moins, c'est ce que Configure pense.

Unterminated <> operator

(F) L'analyseur syntaxique voit un crochet gauche à la place de ce qu'il attendait être un terme, donc il regarde le crochet droit correspondant, et ne le trouve pas. Il y a des chances que vous ayez oublié une parenthèse obligatoire plus tôt dans la même ligne, et vous pensiez vraiment à un « moins que ».

Use of "\$\$<digit>" to mean "\${}\$<digit>" is deprecated

(D) Les versions de Perl antérieures à la 5.004 interprétaient mal n'importe quel type marqué suivi par « \$ » et un chiffre. Par exemple « \$\$0 » est compris incorrectement sous la forme « \${}\$0 » au lieu de « \${0} ». Ce bug est (la plupart du temps) corrigé dans Perl 5.004. Cependant, les développeurs de Perl 5.004 ne peuvent corriger ce bug complètement, car au moins 2 gros modules dépendent de cette ancienne façon de penser que « \$\$0 » est une chaîne. Donc Perl 5.004 continue d'interpréter « \$\$<digit> » de la mauvaise façon dans les chaînes ; mais cela génère un message d'alerte. Et en Perl 5.005, ce traitement spécial cessera.

Use of \$# is deprecated

(D) C'est une tentative échouée d'émuler les possibilités d'un pauvre **awk** Utilisez un explicite `printf()` ou `sprintf()` à la place.

Use of \$* is deprecated

(D) Cette variable magique qui rendait active la recherche d'un motif en multi-lignes, à la fois pour vous et pour les sous-fonctions chanceuses que vous serez amené à appeler. Vous devriez utiliser le nouveau `//m` et `//s` modificateurs pour faire cela sans prendre l'effet d'action-à-distance de `$*`.

Use of %s in printf format not supported

(F) Vous essayez d'utiliser une possibilité de `printf` qui est seulement accessible depuis le C. Cela veut dire habituellement qu'il existe une meilleure manière de faire ça en Perl.

Use of bare << to mean <<"" is deprecated

(D) Vous êtes maintenant encouragé à utiliser l'explicite forme des cotes si vous envisagez d'utiliser une ligne vide comme séparateur d'un here-document.

Use of implicit split to @_ is deprecated

(D) Cela fait beaucoup de travail pour le compilateur quand vous forcez la liste d'arguments d'une sous-fonction, donc il est mieux si vous affectez le résultat d'un `split()` explicitement dans un tableau (ou une liste).

Use of inherited AUTOLOAD for non-method %s() is deprecated

(D) Comme (ahem) dispositif accidentel, l'`AUTOLOAD` des sous-programmes sont recherchés comme des méthodes (utilisant la hiérarchie `@ISA`) même lorsque les sous-programmes à autoloader sont appelés en tant que tout simplement fonctions (par exemple `Foo::bar()`), pas comme méthodes (par exemple `Foo->bar()` ou `$obj->bar()`).

Cette anomalie sera rectifiée dans Perl 5,005, qui utilisera la consultation de méthode seulement pour l'AUTOLOADS des méthodes. Cependant, il y a une base significative de code existant qui peut utiliser le vieux comportement. Ainsi, comme étape d'intérim, Perl 5,004 émet un avertissement facultatif quand l'utilisation de non-méthodes a hérité de AUTOLOAD.

La règle simple est : l'héritage ne fonctionnera pas quand il y a autoloading de non-méthodes. La solution simple pour le vieux code est : Dans tout module qui dépendait d'hériter d'AUTOLOAD pour des non-méthodes d'une classe de base nommée `BaseClass`, exécutez `*AUTOLOAD = &BaseClass::AUTOLOAD` pendant la mise en route.

Dans le code qui dit actuellement `use AutoLoader; @ISA = qw(AutoLoader);` vous pouvez ôter `AutoLoader` de `@ISA` et changer `use AutoLoader;` pour `use AutoLoader 'AUTOLOAD';`.

Use of reserved word "%s" is deprecated

(D) Le mot indiqué est un mot réservé. Les versions futures de Perl peuvent utiliser ce mot comme mot-clef, donc il se serait mieux de coter explicitement le mot d'une manière appropriée pour le contexte, ou d'utiliser un nom différent de toute façon. Ce message peut être supprimé pour les noms de fonction en ajoutant le préfixe `&`, ou en utilisant le qualifiant du paquetage, ex. `&our()`, ou `Foo::our()`.

Use of %s is deprecated

(D) Le constructeur indique qu'il n'est plus recommandé de l'utiliser, généralement parce qu'il y a une meilleure façon de le faire, et parce que l'ancienne méthode a des mauvais effets de bord.

Use of uninitialized value

(W) Une valeur indéfinie a été utilisée comme si elle avait déjà été définie. Elle est interprétée comme un `<"???"` ou `0`, mais c'est peut-être un oubli. Pour supprimer ce message, donnez une valeur initiale à vos variables.

Useless use of "re" pragma

(W) Vous faites `use re;` sans aucun argument. Ce n'est pas vraiment utile.

Useless use of %s in void context

(W) Vous faites quelque chose dans effet de bord `???` dans un contexte qui ne fait rien avec la valeur retournée, comme un état qui ne retourne pas de valeur depuis le block, ou du côté gauche d'un opérateur virgule de scalaire. Très fréquemment ce point n'est pas stupide de votre part, mais c'est un échec de Perl pour parser le programme comme vous voudriez qu'il soit. Par exemple, vous obtenez ceci si vous mixez votre précedence C avec la précedence Python et dites ceci

```
$one, $two = 1, 2;
```

quand vous pensez dire

```
($one, $two) = (1, 2);
```

Une autre erreur commune est d'utiliser les parenthèses ordinaires pour construire une liste de références, lorsque vous pouvez utiliser les accolades ou les crochets, si vous dites

```
$array = (1,2);
```

alors que vous devriez dire

```
$array = [1,2];
```

Les crochets transforment explicitement une liste de valeurs en une valeur scalaire, alors que les parenthèses ne le font pas. Donc quand une liste entre parenthèses est évaluée dans un contexte de scalaire, la virgule est traitée comme l'opérateur C virgule, ce que qui `???` rejette l'argument de gauche, ce qui n'est pas ce que vous voulez. Voir *perlref* pour plus de détails sur ceci.

untie attempted while %d inner references still exist

(W) Une copie de l'objet retourné depuis `tie` (or `tied`) est toujours valide quand `untie` est appelé.

Value of %s can be "0"; test with defined()

(W) Dans une expression conditionnelle, vous utilisez `<DESCRIPTEUR>`, `<*>` (`glob`), `each()`, ou `readdir()` comme une valeur booléenne. Chacun de ces constructeurs peut retourner une valeur de `< 0 >`; Cela va rendre l'expression conditionnelle fausse, ce qui n'est probablement pas ce que vous désirez. Quand vous utilisez ces constructeurs dans une expression conditionnelle, testez leur valeur avec l'opérateur `defined`.

Variable "%s" is not imported %s

(F) Alors que `< use strict >` est en action, vous faites référence à une variable globale que vous pensez avoir importée d'un autre module, parce que quelque chose ayant le même nom (habituellement une sous-fonction) est exportée par ce module. Cela veut dire habituellement que vous avez mis un mauvais caractère devant votre variable.

Variable "%s" may be unavailable

(W) Une fonction interne *anonyme* est dans une fonction *nommée*, et en dehors de cela il y a une autre fonction ; et la fonction anonyme (la plus à l'intérieur) fait référence à une variable lexicale définie dans la fonction la plus à l'extérieur. Par exemple :

```
sub laplusalexterieur { my $a; sub aumilieu { sub { $a } } }
```

Si la sous-fonction anonyme est appelée ou référencée (directement ou indirectement) depuis la fonction la plus à l'extérieur, elle va partager la variable comme vous le souhaitez. Mais si la sous-fonction anonyme est appelée ou référencée quand la fonction la plus à l'extérieur est inactive, elle verra la valeur de la variable partagée comme elle était avant et durant le **premier** appel à la fonction la plus à l'extérieur, ce qui n'est probablement pas ce que vous voulez.

Dans ces circonstances, il est habituellement mieux de faire la fonction du milieu anonyme, en utilisant la syntaxe `sub {}`. Perl a des supports spécifiques pour les variables partagées dans les fonctions anonymes internes ; une fonction nommée entre les deux interfère avec cette possibilité.

Variable "%s" will not stay shared

(W) Une sous-fonction interne *nommée* référence une variable lexicale définie dans une routine externe.

Quand la sous-fonction interne est appelée, elle va probablement voir la valeur de la variable de la fonction externe comme elle était avant et durant le **premier** appel à la sous-fonction externe ; Dans ce cas, après que le premier appel à la sous-fonction externe soit fini, les sous-fonctions ne partageront plus une valeur commune pour la variable. En d'autres termes, la variable ne sera plus partagée.

De plus, si la sous-fonction externe est anonyme et référence une variable lexicale en dehors d'elle-même, alors les sous-fonctions interne et externe ne vont *jamais* partager la variable donnée.

Le problème peut habituellement être résolu en faisant la sous-fonction interne anonyme, en utilisant la syntaxe `sub {}`. Quand les fonctions internes qui référencent des variables dans les sous-fonctions externes sont appelées ou référencées, elles font automatiquement rebondir vers la valeur courante de chacune des variables.

Variable syntax

(A) Vous lancez accidentellement votre script par **csh** au lieu de Perl. Vérifiez la ligne avec **#!**, ou lancez votre script manuellement dans Perl.

perl: warning: Setting locale failed.

(S) Les message entiers ressemblent à ça :

```
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LC_ALL = "En_US",
    LANG = (unset)
are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
```

Exactement ce qui se passe lorsque le changement de configuration local échoue. Dans l'exemple, la configuration locale indique que `LC_ALL` est à « `En_US` » et que `LANG` n'a pas de valeur. Cette erreur indique que Perl a détecté que vous et/ou votre administrateur système ont configuré les soi-disant variables systèmes mais Perl ne peut utiliser cette configuration. Ce n'est pas très grave heureusement : Il existe une « default locale » appelée « `C` » que Perl peut et va utiliser, et le script va marcher. Mais avant que vous ne corrigiez ce problème, vous allez avoir le même message d'erreur à chaque fois que vous lancerez Perl. Comment vraiment corriger le problème est indiqué dans *perllocale* section **LOCALE PROBLEMS**.

Warning: something's wrong

(W) Vous passez à `warn()` une chaîne vide (l'équivalent d'un `warn ""`) ou vous l'appellez sans arguments et `$_` est vide.

Warning: unable to close filehandle %s properly

(S) L'appel implicite à `close()` fait par `open()` donne une erreur indiquée dans le `close()`. Cela indique habituellement que votre système n'a plus d'espace disque.

Warning: Use of "%s" without parentheses is ambiguous

(S) Vous avez écrit un opérateur unaire suivi par quelque chose qui ressemble à un opérateur unaire qui peut être interprété comme un terme ou un opérateur unaire. Pour exemple, si vous savez que la fonction `rand` a un argument par défaut de 1.0, et que vous écrivez

```
rand + 5;
```

Vous PENSEZ que vous écrivez la même chose que

```
rand() + 5;
```

alors qu'en fait vous obtenez

```
rand(+5);
```

Donc mettez des parenthèses pour dire ce que vous pensez vraiment.

Write on closed filehandle

(W) Le descripteur de fichier que vous écrivez a été fermé quelque temps avant. Vérifiez votre flux logique de données.

X outside of string

(F) Vous avez un pack template qui spécifie une position relative avant le début de la chaîne qui est en train d'être décompactée. Voir `pack()` dans *perlfunc*.

x outside of string

(F) Vous avez un pack template qui spécifie une position relative après la fin de la chaîne qui est en train d'être décompactée. Voir `pack()` dans *perlfunc*.

Xsub "%s" called in sort

(F) L'utilisation d'un sous-programme comme comparaison avec `sort` n'est pas encore gérée.

Xsub called in sort

(F) L'utilisation d'un sous-programme comme comparaison avec `sort` n'est pas encore supporté.

You can't use -1 on a filehandle

(F) Un descripteur de fichier représente un fichier ouvert, et quand vous ouvrez le fichier, il a déjà passé tous les liens symboliques que vous présumiez essayer de regarder. Utilisez un nom de fichier plutôt.

YOU HAVEN'T DISABLED SET-ID SCRIPTS IN THE KERNEL YET!

(F) Et ce n'est probablement jamais ce que vous avez voulu, parce que vous n'avez pas les sources du kernel, et que votre vendeur ne vous donnera rien de ce que vous voulez. La meilleure solution est d'utiliser le script `wrapsuid` dans le répertoire pour mettre une couche C `setuid` autour de votre script.

You need to quote "%s"

(W) Vous avez affecté un mot comme nom d'un gestionnaire de signaux. Malheureusement, vous avez déjà un sous-programme avec ce nom, ce qui veut dire que Perl 5 va essayer ce sous-programme quand l'affectation va être exécutée. (Si c'est que vous voulez, mettre un `&` devant.)

[gs]setsockopt() on closed fd

(W) Vous essayez de lire ou de positionner une option sur une socket fermée. Peut-être avez-vous oublié de vérifier la valeur retournée par l'appel de `socket()`? Voir `getsockopt()` dans *perlfunc*.

\1 better written as \$1

(W) En dehors des modèles ('patterns'), les références inversées fonctionnent comme des variables. L'utilisation de barres obliques inverses est historiquement du côté droit d'une substitution, mais stylistiquement???? il est mieux d'utiliser la forme variable car les autres programmeurs Perl s'y attendent, et cela marche mieux s'il existe plus de 9 références inversées.

'<' and '<' may not both be specified on command line

(F) Une erreur spécifique à VMS. Perl fait ses propres redirections de ligne de commandes, et il a trouvé que l'entrée standard est un pipe, et que vous essayez de rediriger STDIN en utilisant '<'. Seulement un flux sur l'entrée standard pour un client, s'il vous plaît.

'>' and '>' may not both be specified on command line

(F) Une erreur spécifique à VMS. Perl fait ses propres redirections de ligne de commandes, et il pense que vous essayez de rediriger la sortie standard à la fois dans un fichier `and` dans un pipe vers une autre commande. Vous devez choisir l'un ou l'autre, en sachant que rien ne vous empêche de faire un pipe dans un programme ou dans un script Perl qui 'coupe' la sortie en deux flux, comme

```
open(OUT, ">$ARGV[0]") or die "Can't write to $ARGV[0]: $!";
while (<STDIN>) {
    print;
    print OUT;
}
close OUT;
```

Got an error from DosAllocMem

(P) Une erreur spécifique à OS/2. Le plus probable est que vous utilisez une version obsolète de Perl, ce qui ne devrait pas arriver.

Malformed PERLLIB_PREFIX

(F) Une erreur spécifique à OS/2. PERLLIB_PREFIX doit être de la forme

```
prefix1;prefix2
```

ou

```
prefix1 prefix2
```

avec `prefix1` et `prefix2` non vides. Si `prefix1` est un préfixe d'une librairie cherchée dans le path, `prefix2` est substitué. L'erreur peut apparaître si un composant n'est pas trouvé, ou trop long. Voir « PERLLIB_PREFIX » dans *README.os2*.

PERL_SH_DIR too long

(F) Erreur particulière à OS/2. PERL_SH_DIR est le répertoire où se trouve le shell `sh-shell`. Voir « PERL_SH_DIR » dans *README.os2*.

Process terminated by SIG%s

(W) C'est un message standard formulé par les applications OS/2, alors que les applications *nix se terminent en silence. C'est considéré comme une caractéristique du portage sous OS/2. Ceci peut être facilement désactivé en positionnant le gestionnaire de signal approprié, voir Signaux in *perlipc*. Voir aussi « Process terminated by SIGTERM/SIGINT » dans *README.os2*.

3 TRADUCTION

3.1 Version

Cette traduction française correspond à la version anglaise distribuée avec perl 5.005_02. Pour en savoir plus concernant ces traductions, consultez <http://perl.enstimac.fr/>.

3.2 Traducteurs

Alain Barbet <abarb@nmg.fr>

3.3 Relecture

Gérard et Inès Delafond.

4 À propos de ce document

Ce document est la traduction française du document original distribué avec perl. Vous pouvez retrouver l'ensemble de la documentation française Perl (éventuellement mise à jour) en consultant l'URL <<http://perl.enstimac.fr/>>.

Ce document PDF a été produit Paul Gaborit. Si vous utilisez la version PDF de cette documentation (ou une version papier issue de la version PDF) pour tout autre usage qu'un usage personnel, je vous serai reconnaissant de m'en informer par un petit message <<mailto:Paul.Gaborit@enstimac.fr>>.

Si vous avez des remarques concernant ce document, en premier lieu, contactez la traducteur (vous devriez trouver son adresse électronique dans la rubrique TRADUCTION) et expliquez-lui gentiment vos remarques ou critiques. Il devrait normalement vous répondre et prendre en compte votre avis. En l'absence de réponse, vous pouvez éventuellement me contacter.

Vous pouvez aussi participer à l'effort de traduction de la documentation Perl. Toutes les bonnes volontés sont les bienvenues. Vous devriez trouver tous les renseignements nécessaires en consultant l'URL ci-dessus.

Ce document PDF est distribué selon les termes de la license Artistique de Perl. Toute autre distribution de ce fichier ou de ses dérivés impose qu'un arrangement soit fait avec le(s) propriétaire(s) des droits. Ces droits appartiennent aux auteurs du document original (lorsqu'ils sont identifiés dans la rubrique AUTEUR), aux traducteurs et relecteurs pour la version française et à moi-même pour la version PDF.