Package 'tools'

September 21, 2016

Version 3.3.1

Priority base

Title Tools for Package Development

Author R Core Team

Maintainer R Core Team <R-core@r-project.org>

Description Tools for package development, administration and documentation.

License Part of R 3.3.1

Suggests codetools, methods, xml2

R topics documented:

tools-package . . . 2
.print.via.format . . . 3
add_datalist . . . 4
assertCondition . . . 4
bibstyle . . . 6
buildVignette . . . 8
buildVignettes . . . 9
charsets . . . 10
checkFF . . . 11
checkMD5sums . . . 13
checkPoFiles . . . 13
checkRd . . . 15
checkRdaFiles . . . 17
checkTnF . . . 18
checkVignettes . . . 19
check_packages_in_dir . . . 20
codoc . . . 22
compactPDF . . . 24
delimMatch . . . 26
dependsOnPkgs . . . 26
encoded_text_to_latex . . . 27
fileutils . . . 29
find_gs_cmd . . . 30
getDepList . . . 31
getVignetteInfo . . . 32
HTMLheader . . . 33

HTMLlinks . . . 34
installFoundDepends . . . 35
loadRdMacros . . . 36
makevars . . . 37
make_translations_pkg . . . 38
md5sum . . . 38
package.dependencies . . . 39
package_dependencies . . . 40
parseLatex . . . 41
parse_Rd . . . 42
pskill . . . 44
psnice . . . 45
QC . . . 46
Rcmd . . . 47
Rd2HTML . . . 48
Rd2txt_options . . . 51
Rdiff . . . 52
Rdindex . . . 53
RdTextFilter . . . 54
Rdutils . . . 55
read.00Index . . . 56
showNonASCII . . . 56
startDynamicHelp . . . 57
SweaveTeXFilter . . . 58
testInstalledPackage . . . 59
texi2dvi . . . 60
toHTML . . . 62
tools-deprecated . . . 63
toRd . . . 63
toTitleCase . . . 64
undoc . . . 64
update_pkg_po . . . 65
vignetteDepends . . . 67
vignetteEngine . . . 68
write_PACKAGES . . . 69
xgettext . . . 71

Index **73**

`tools-package` パッケージ開発のためのツール

Description

パッケージ開発，管理そしてドキュメント化のためのツール.

Details

このパッケージは R パッケージの操作とそれらのドキュメントのためのツールを含む.
関数の完全なリストには `library(help = "tools")` を使う.

Author(s)

Kurt Hornik と Friedrich Leisch

保守管理者：R Core Team <R-core@r-project.org>

`.print.via.format` プリント用ユーティリティ

Description

`.print.via.format` は `print()` メソッドの “プロトタイプ”で，少なくとも次のように簡単な出発点として有用である

```
print.<myS3class>  <-  .print.via.format
```

Usage

```
.print.via.format(x, ...)
```

Arguments

`x`	プリントされるオブジェクト.
`...`	オプションの追加引数で `format` に渡される.

Value

`x`, `print` メソッドがそうあるべきであるように不可視 (`invisible()` を用いる).

See Also

総称的関数 `print`；その既定メソッドは `print.default` (`"numeric"`, `"character"` そしてそれらの `array`， codelists 等の多くの基本暗黙クラスに対して使われる).

Examples

```
## この関数は単純に次のように定義される
 function (x, ...) {
    writeLines(format(x, ...))
    invisible(x)
 }

## R の簡単なプリントメソッドと新しいメソッドのプロトタイプして使われる.
```

add_datalist　　パッケージに 'datalist' ファイルを加える

Description

引数無しの data() は付加されたパッケージの data を使い利用可能なデータセットのリストを表示し，そのためにパッケージごとのリストがインストールされる．このパッケージをインストール時に作るのは膨大なデータセットを持つパッケージに対しては遅い可能性があり，'data/datalist' ファイルを提供することで迅速に出来る．

Usage

```
add_datalist(pkgpath, force = FALSE)
```

Arguments

pkgpath	(ソース)パッケージへのパス．
force	論理値：既存の 'data/datalist' ファイルを上書きすべきか?

Details

R CMD build は 1MB 又はそれ以上のデータを持つパッケージにデータリストを付け加えるためにこの関数を呼び出す．

データセットが多くの従属性を持つファイルに 'data/datalist' を与えることはまた，パッケージ自体(そして恐らくその他)のロードを含み有用である．

See Also

data.

マニュアル 'Writing R Extensions'．

assertCondition　　エラー条件を確認

Description

コードのテストでは，結果が正しいことをチェックするだけでは不十分で，同時にエラーや警告が適当な状況でシグナルされることが必要である．ここで説明される関数はそれを行う便利な機能を提供する．この三つの関数は提供された表現式の評価がエラー，警告，又は指定された条件のリストの一つを生成するかどうかをそれぞれチェックする．もし主張が失敗すれば警告が出る．

Usage

```
assertError(expr, verbose = FALSE)
assertWarning(expr, verbose = FALSE)
assertCondition(expr, ..., .exprString = , verbose = FALSE)
```

Arguments

expr	tryCatch(expr, ..) で評価される未評価の R 表現式.
...	主張を満足する条件のクラスに対応する文字列，例えば "error" や "warning". もし何も指定されなければ，任意の条件は主張を満足する. 詳細節を見よ.
.exprString	expr に対応するプリントされる文字列. 既定では，実際の expr が逆構文解析される. もし関数がテストされる実際の表現式と共に提供されると省略される. もし assertCondition() が別の関数から呼び出され，実際の表現式がその関数への引数として渡されると，逆構文解析されたバージョンを提供する.
verbose	もし TRUE ならば，条件が満足されるとメッセージがプリントされる.

Details

assertCondition() は expr の評価中で生成される全ての条件をチェックするために一般的な条件機構を使う. これらの中の提供された条件クラスのいずれかの生起は他の条件が何をシグナルしようと主張を満足する.

assertError() はエラーを主張するための簡便関数である；これは assertCondition() を呼び出す.

assertWarning() は警告が出ることを主張するが，エラーはしない，他方で assertCondition(expr, "warning") は警告にエラーが続いても満足される. 例を見よ.

Value

もし主張が満足されると，シグナルを出した全ての条件のリストが不可視で返される. これらのオブジェクトの解釈は conditionMessage を見よ. 評価中にシグナルを出した全ての条件が，それらが要求中に有るか無いかを問わず返されることを注意する.

Author(s)

John Chambers と Martin Maechler

See Also

stop, warning; signalCondition, tryCatch.

Examples

```
  assertError(sqrt("abc"))
  assertWarning(matrix(1:8, 4,3))

  assertCondition( ""-1 ) # ok, 全ての条件はこれを満足する

try( assertCondition(sqrt(2), "warning") )
## .. sqrt(2) の評価で警告を得ることに失敗
     assertCondition(sqrt("abc"), "error")   # ok
try( assertCondition(sqrt("abc"), "warning") )# -> error: 警告が出ない
     assertCondition(sqrt("abc"), "error")
  ## 上の assertError() 呼び出しと同一

assertCondition(matrix(1:5, 2,3), "warning")
try( assertCondition(matrix(1:8, 4,3), "error") )
## .. 期待されるエラーを得ることに失敗 ....
```

```
## 警告又はもっとまずいこと：
assertCondition(matrix(1:8, 4,3), "error","warning") # OK
assertCondition(matrix(1:8, 4, 3), "warning") # OK

## 両方がシグナルされる時：
ff <- function() { warning("my warning"); stop("my error") }
    assertCondition(ff(), "warning")
## しかし assertWarning はエラーが続くことを許さない
try(assertWarning(ff()))
    assertCondition(ff(), "error")          # ok
assertCondition(ff(), "error", "warning") # ok (黙って，警告をキャッチ)

## assertC..() が主張しないことを主張する[そして*一つ*の引数だけ]
assertCondition( assertCondition(sqrt( 2   ), "warning") )
assertCondition( assertCondition(sqrt("abc"), "warning"), "error")
assertCondition( assertCondition(matrix(1:8, 4,3), "error"),
                "error")
```

---

bibstyle　　文献スタイルを選択，定義する

Description

この関数は bibentry オブジェクトを，後でテキストや HTML 等に変換するために定義したり登録したりする．

Usage

```
bibstyle(style, envir, ..., .init = FALSE, .default = TRUE)
getBibstyle(all = FALSE)
```

Arguments

style	スタイルの名前を与える文字列．
envir	(オプションの)スタイルを実装する関数を保持する環境．
...	環境に付け加えられる名前付き引数．
.init	既定のスタイル "JSS" から環境を初期化するか．
.default	指定したスタイルを既定スタイルに設定するか．
all	全ての登録されたスタイルの名前を返すか．

Details

bibentry オブジェクトの翻訳は BibTeX で使われているようなものに倣ってモデル化されたルーチンを用いてなされる．この関数はこうしたルーチンを含む環境を作り操作することを許す．

新しいスタイル環境を作る二つの方法が有る．最も簡単なものは .init = TRUE を設定することで，この場合環境は既定の "JSS" 環境のコピーで初期化される．(このスタイルは *Journal of Statistical Software* で使われている 'jss.bst' スタイルに倣ってモデル化されている．) 別の方法としては，全く新しいスタイル環境を指定するのに envir 引数を使うことが出来る．

既定のスタイルの名前を見つけるには getBibstyle() を使う．既存のスタイルを既定に設定することなしに検索するには， bibstyle(style, .default = FALSE) を使う．既存のスタイルを改変するには， style を指定しある名前付き項目を ... を使って指定する．(既定の "JSS" スタイルの改変は勧められない．) style を NULL に設定するか，又はそれを欠損のままにすると既定のスタイルが回収されるが，改変は許されない．

最低でも環境は bibentry によりサポートされる12の文献項目タイプの各々と各々と並び以下で説明される幾つかの他のルーチンのを翻訳するルーチンを含むべきである．前者は次のような名前を持たなければならない； formatArticle, formatBook, formatInbook, formatIncollection, formatInProceedings, formatManual, formatMastersthesis, formatMisc, formatPhdthesis, formatProceedings, formatTechreport そして formatUnpublished. これらの各々は一つの引数，翻訳器に渡された bibentry ベクトルからの単一のクラスを持たない項目，を取り，そして単一の要素を持つ文字列ベクトルを生成すべきである (新行文字を含むかもしれない).

他のルーチンは次のようである．項目をソートするソートキーを作り出す sortKeys はオリジナルの bibentry ベクトルを渡しソート順序を定義する同じ長さのソート可能なベクトルを生成する．最後に，オプションの関数 cite は utils::cite と同じ引数リストを持つべきであり，そしてテキスト中で使われる引用文献を作り出すべきである．

"bibentry" オブジェクトに対する format メソッドはソート後で整形前に各項目に ".index" という名前の欄を加える．これは番号付けを必要とするスタイル中で使える完全なオブジェクト中の 1-based 索引である．"JSS" スタイルは番号を使わないが，それらを表示するのに使うことができるスタブ関数 fmtPrefix() を含む．下の例を見よ．

Value

bibstyle は選択されたか作られた環境を返す．

getBibstyle は既定のスタイルの名前を返す，又は全てのスタイル名を返す．

Author(s)

Duncan Murdoch

See Also

bibentry

Examples

```
refs <-
c(bibentry(bibtype = "manual",
    title = "R: A Language and Environment for Statistical Computing",
    author = person("R Core Team"),
    organization = "R Foundation for Statistical Computing",
    address = "Vienna, Austria",
    year = 2013,
    url = "https://www.R-project.org"),
  bibentry(bibtype = "article",
    author = c(person(c("George", "E.", "P."), "Box"),
               person(c("David",  "R."),      "Cox")),
    year = 1964,
    title = "An Analysis of Transformations",
    journal = "Journal of the Royal Statistical Society, Series B",
    volume = 26,
```

```
    pages = "211-252"))

bibstyle("unsorted", sortKeys = function(refs) seq_along(refs),
    fmtPrefix = function(paper) paste0("[", paper$.index, "]"),
       .init = TRUE)
print(refs, .bibstyle = "unsorted")
```

---

buildVignette　　ビニエットを作る

Description

一つのビニエットに対して Sweave (または他の weave 関数) texi2dvi そして/または Stangle (または他のカスタム tangle 関数)を実行する.

これは R CMD Sweave の実働関数である.

Usage

```
buildVignette(file, dir = ".", weave = TRUE, latex = TRUE, tangle = TRUE,
              quiet = TRUE, clean = TRUE, keep = character(),
              engine = NULL, buildPkg = NULL, encoding, ...)
```

Arguments

file	文字列；ビニエットのソースファイル
dir	文字列；中間ファイルと出力ファイルが作られる作業ディレクトリ
weave	論理値；weave を実行するか?
latex	論理値；weave が '.tex' ファイルを作ったならば texi2pdf を実行するか?
tangle	論理値；tangle を実行するか?
quiet	論理値；静音モードで実行するか?
clean	論理値；しばしば中間であるある種の新しく作られたファイルを取り除くかどうか. 下の詳細節を見よ.
keep	掃除の際にいかなる場合も保存して置くファイル名のリスト. "target" ファイルはいずれにせよ保存されることを注意する.
engine	NULL 又は文字列；使用するビニエットエンジンの名前. ビニエット中の任意の \VignetteEngine{} マークアップに優先する.
buildPkg	NULL 又は文字列；その中でビニエットエンジンをみつけるオプションのパッケージ.
encoding	ファイルに対して仮定されたエンコーディング. もし指定されないとそれはファイルの中身から推測される.
...	weave と tangle に渡される追加引数.

Details

この関数はビニエットに対するビニエットエンジンを決定し (既定は utils::Sweave), それからそのエンジンを使ってビニエットを weave したり(と) tangle を実行する．最後にもし clean が TRUE ならば，新しく作られた中間ファイル (エンジン等に依存する“ターゲット”でなく，そして keep 中に有るどれでもなく) は削除される．もし clean が NA で weave が真なら，新しく作られた中間出力ファイル (例えば ‘.tex’) はたとえそれらから ‘.pdf’ ファイルが既に作られていても消されない．

もし buildPkg が指定されると，それはビニエットが処理される前にロードされ，ビニエットエンジンの探索中に既定パッケージとして使われるが，ビニエットのソース中で明示的に指定されたパッケージ (例えば **utils** パッケージ中の Sweave エンジンを指定するのに \VignetteEngine{utils::Sweave} を使う)はそれに優先する．反対にもし engine 引数が与えられると，それはビニエットソースに優先する．

Value

作られたファイルの名前である文字列ベクトル．

Author(s)

Henrik Bengtsson と Duncan Murdoch

See Also

パッケージ中の全てのビニエットを構築するには buildVignettes.

---

buildVignettes　　パッケージのビニエットをリスト，構築する

Description

パッケージの全てのビニエットに対して Sweave (又は他のカスタム weave 関数) と texi2dvi を実行する．

Usage

```
buildVignettes(package, dir, lib.loc = NULL, quiet = TRUE,
               clean = TRUE, tangle = FALSE)

pkgVignettes(package, dir, subdirs = NULL, lib.loc = NULL,
     output = FALSE, source = FALSE, check = FALSE)
```

Arguments

package	インストールされたパッケージの名前を与える文字列．もし与えられると既定で副ディレクトリ ‘doc’ 中でビニエットもソースファイルが探される．
dir	パッケージのルートソースディレクトリへのパスを指定する文字列．この副ディレクトリ ‘vignettes’ (又はそれが無ければ ‘inst/doc’) でビニエットのソースファイルが探される．

lib.loc	R ライブラリのディレクトリ名の文字列ベクトル，又は NULL．既定値 NULL は現在知られている全てのライブラリに対応する．指定されたライブラリツリーは package を探すのに使われる．
quiet	論理値．weave と texi2pdf を沈黙モードで実行する．
clean	構築で作られた全てのファイルをそこに前からのコピーがあっても削除する．
tangle	論理値．weave だけでなく tangle も実行するか．
subdirs	ビニエットをその中で探す dir の副ディレクトリの文字列ベクトル．存在する最初のものが使われる．package が提供されると既定では "doc"，さもなければ "vignettes"．
output	各ビニエットの出力ファイル名を返すべきかどうかを指示する論理値 (成分 outputs 中に)．
source	各ビニエットに対する *tangle* された出力を返すべきか(成分 sources 中に)．
check	論理値．もし TRUE ならば，ビニエット風のファイル名を持つ全てのファイルが特定できるビニエットエンジンを持つかどうかをチェックする．もしファイルがビニエットでないがビニエットエンジンのどれかで定義されたパターンにマッチするファイル名を持てば，誤って報告されるかもしれない．

Details

buildVignettes はビニエットの PDF をそれらのソースから(再)構築するために R CMD build と R CMD check により使われる．

Value

buildVignettes は全てのビニエットの PDF 版を作るというその副作用のために呼び出され，もし tangle = TRUE ならば R コードを取り出す．

pkgVignettes はもしビニエットディレクトリがあればクラス "pkgVignettes" のオブジェクトを，さもなければ NULL を返す．

Examples

```
gVigns <- pkgVignettes("grid")
str(gVigns)
```

---

charsets 文字セット間の変換表

Description

charset_to_Unicode は Unicode のコードポイントの行列で通常の8ビットエンコーディングに対する列を持つ．

Adobe_glyphs は Unicode のコードポイントに対する Adobe グリフ名を与えるデータフレームである．これは二つの文字列の列 "adobe" と "unicode" (4桁の16進表現)を持つ．

Usage

```
charset_to_Unicode

Adobe_glyphs
```

Details

charset_to_Unicode はクラス c("noquote", "hexmode") の整数行列であり，従って16進法でプリントされる．マッピングは libiconv で使われるものである：引用符とマイナス/ハイフンがマップされる仕方にソース間で食い違いが有る (そしてポストスクリプトのエンコーディングファイルは異なったマッピングを使う).

Adobe_glyphs は全ての Adobe グリフ名を含み単一の Unicode 文字に対応する．これは Unicode コードポイントでソートされており，ポイント内ではグリフのアルファベット順である (Unicode コードポイントには複数の名前があり得る)．データは次のファイル内に有る '*R_HOME*/share/encodings/Adobe_glyphlist'

Source

https://partners.adobe.com/public/developer/en/opentype/glyphlist.txt

Examples

```
## ISOLatin2 文字に対する Adobe 名を与える
latin2 <- charset_to_Unicode[, "ISOLatin2"]
aUnicode <- as.numeric(paste0("0x", Adobe_glyphs$unicode))
keep <- aUnicode %in% latin2
aUnicode <- aUnicode[keep]
aAdobe <- Adobe_glyphs[keep, 1]
## 最初のマッチ
aLatin2 <- aAdobe[match(latin2, aUnicode)]
## 全てのマッチ
bLatin2 <- lapply(1:256, function(x) aAdobe[aUnicode == latin2[x]])
format(bLatin2, justify = "none")
```

---

checkFF 外部言語関数呼び出しのチェック

Description

R こーどからのコンパイルコードへの呼び出しに関するチェックを行う．現在 .C と .Fortran の様なインタフェイス関数が最初の引数 "NativeSymbolInfo" 又は引数 PACKAGE が指定されて呼び出されているかどうかをチェックする．これは外部言語呼び出し中の名前の衝突を避けるため是非とも勧められる．

Usage

```
checkFF(package, dir, file, lib.loc = NULL,
        registration = FALSE, check_DUP = FALSE,
        verbose = getOption("verbose"))
```

Arguments

package	インストールされたパッケージの名前を与える文字列．もし与えられると，パッケージのインストールされた R がチェックされる．
dir	パッケージのルートソースディレクトリへのパスを指定する文字列．これは副ディレクトリ 'R' (R コード用)を含むべきである． package が与えられない時だけ使われる．
file	チェックされるべき R コードを含むファイル名． package も dir も与えられない時使われる．
lib.loc	R ライブラリのディレクトリ名の文字列，又は NULL．既定値の NULL は現在知られている全てのライブラリに対応する．指定されたライブラリツリーは package を探すのに使われる．
registration	論理値．もし TRUE なら呼び出しに関する登録情報をチェックする (もし利用できれば)．
check_DUP	論理値．もし TRUE なら， DUP = FALSE である .C と .Fortran 呼び出しが報告される．
verbose	論理値．もし TRUE なら，追加の診断がプリントされる (そして結果は不可視で返される)．

Details

name 引数がシンボルか文字列であることだけがチェックされ，シンボルが実行時にどんなオブジェクトのクラスに解決されるかはチェックされないことを注意する．

もしパッケージがディレクティブ useDynLib を含む名前空間を持てば，パッケージ中のトップレベル関数中の呼び出しは報告されない．それらのシンボルは最初の useDynLib ディレクティブ中で命名された DLL 中で優先的に探されるからである．

これは指定された PACKAGE の呼び出しが同じパッケージに対するものであることをチェックし，そして基本パッケージ中に有るものと他のパッケージ中に有るものを個別に報告する (そしてもしこれらのパッケージが 'DESCRIPTION' ファイル中で指定されていれば)．

Value

クラス "checkFF" のオブジェクト．

そうしたオブジェクト中の情報を表示するメソッド format と print が有る．

See Also

.C, .Fortran; Foreign.

Examples

```
 # 順序はかなりランダム
checkFF(package = "stats", verbose = TRUE)
```

checkMD5sums *MD5* チェックサムファイルをチェック，生成する

Description

checkMD5sums はファイルを 'MD5' ファイルに対してチェックする．

Usage

```
checkMD5sums(package, dir)
```

Arguments

package	インストールされたパッケージの名前．
dir	インストールされたパッケージのトップレベルディレクトリへのパス．

Details

ファイル 'MD5' はもし適当なコマンドラインのバージョンの md5sum が利用できればチェック md5sum -c MD5 が可能な書式で作られる． (Windows に対してはそうした一つがバンドル https://cran.r-project.org/bin/windows/Rtools 中に有る．)

もし dir が欠損していれば，名前 package のインストール済みのパッケージが探される．

プライベットな関数 tools:::.installMD5sums が Windows ビルド中の MD5 ファイルを作るために使われる．

Value

checkMD5sums は論理値を返す．もしチェックすべき 'MD5' ファイルがなければ NA を返す．

See Also

md5sum

---

checkPoFiles 一貫性のない書式文字列に対する翻訳ファイルのチェック

Description

これらの関数は一貫性のチェックのために英文メッセージに埋め込まれた書式を翻訳文字列と比較する． checkPoFile は一つのファイルをチェックする一方， checkPoFiles は指定された言語に対する全てのファイルをチェックする．

Usage

```
checkPoFile(f, strictPlural = FALSE)
checkPoFiles(language, dir = ".")
```

Arguments

f	単一のファイルパスを与える文字列.
strictPlural	単数と複数の書式を厳格に非アックするかどうか.
language	言語コードを与える文字列.
dir	ファイルをその中でチェックするディレクトリへのパス.

Details

R の国際化の一部は '.po' ファイル中のメッセージの翻訳に依存している. これらのファイル中には R のソースから取られた'英語'のメッセージに別の言語への翻訳が続く. これらのメッセージの多くは C や R の sprintf と関連関数の書式文字である. これらのケースでは翻訳は一貫性のある書式を与える必要があり, さもなければメッセージが表示される時エラーが起きる.

一貫性の規則は幾つかの点で C と R では異なる. C は R がサポートしていない幾つかの変換をサポートしている, つまり c, u, p, n. C の sprintf() 関数では書式文字列が必要とするよりも多くの引数を持つことが許されているが, R では数は正確に一致しなければならない. R は引数のタイプがマッチすることを要求するが, C はそれが意味があろうとなかろうと表示を行う.

これらの関数はテストに対して次のように妥協する. C で許される全ての書式は受け入れられ, 引数のタイプと数の違いは全て報告される. 結果として報告されるある種の違いはエラーではない.

もし strictPlural 引数が TRUE ならば, 引数リストはメッセージの単数形と複数形の間で正確に一致しなければならない ; もし FALSE ならば, 翻訳は二つの形式の一つかもうひとつにマッチしさえすれば良い. checkPoFiles は checkPoFile を呼び出す時, strictPlural 引数は 'R-' で始まる名前を持つファイルに対して TRUE に設定され, さもなければ FALSE に設定される.

'.po' ファイル中で 'fuzzy' とマークされた項目は処理されない(それらはメッセージコンパイラにより無視されるから).

もし食い違いが見つかると, 翻訳文字列が変形パーセント符号 (例えば幅広パーセント符号 "\uFF05")に対してチェックされる. そうした符号は書式指定子として認識されず, エラーになる可能性がある.

Value

どちらの関数も S3 クラス "check_po_files" のオブジェクトを返す. このクラスに対しては食い違いに関する報告を表示する print メソッドがある.

Author(s)

Duncan Murdoch

References

'.po' ファイルの書式については GNU の gettext のマニュアルを見よ :
https://www.gnu.org/software/gettext/manual/gettext.html.

See Also

update_pkg_po() which calls checkPoFile(); xgettext, sprintf.

Examples

```
## Not run:
checkPoFiles("de", "/path/to/R/src/directory")

## End(Not run)
```

---

checkRd *Rd* オブジェクトをチェックする

Description

ヘルプファイル又は parse_Rd 関数の出力をチェックする.

Usage

```
checkRd(Rd, defines = .Platform$OS.type, stages = "render",
        unknownOK = TRUE, listOK = TRUE, ..., def_enc = FALSE)
```

Arguments

Rd	入力として使うファイル名又は Rd オブジェクト.
defines	#ifdef テスト中で使う文字列.
stages	どの段階("build", "install" 又は "render") で \Sexpr マクロを実行すべきか? 下の注意を見よ.
unknownOK	認識できないマクロは FALSE であるかのようにしてエラーとして，さもなければ警告として扱う.
listOK	不必要な空でない括弧(例えばテキストまわり，引数としてではない)は FALSE であるかのようにエラーとして，さもなければ警告として扱う.
...	Rd がファイル名の時 parse_Rd に渡される追加引数. しばしば有用なものは encoding.
def_enc	論理値：パッケージはエンコーディングを宣言されており，従って非ASCIIテキストは抑制されるか?

Details

checkRd は Rd ファイルに関する一貫性のチェックを行い，必要とされるセクションが存在するかなどを確認する.

これは Rd ファイルに対するファイル名を受け入れ，チェックの実行前にそれを構文解析するために parse_Rd を使う. もしそうなら parse_Rd からの警告が内部関数 prepare_Rd からのものと一緒に集められる. 内部関数は #ifdef と \Sexpr の処理を行い，翻訳出来ないセクションや重複(そしてあってはならない)セクションを取り去り，空のセクションを削除する.

Rd オブジェクトは prepare_Rd を経由して渡されるが，既にそうされているかもしれない (そしてインストールされた Rd オブジェクトはそうである).

警告は'レベル'を与えられる：prepare_Rd からのものはレベル 0 である. これらは以下を含む

全てのテキストはあるセクション中になければならない
一つの *tag name* セクション名だけが許される：最初のものが使われる

セクション *name* は認識されず捨て去られる
空のセクション *name* の除去

checkRd 自体は以下を示すことが出来る

7 タグ *tag name* は認識できない
7 \tabular 書式は簡単でなければならない
7 認識できない \tabular 書式：...
7 このテーブルには *n* 列だけが許される
7 *tag name* を持たねばならない
7 一つの *tag name* だけが許される
7 タグ *tag name* は空であってはならない
7 \docType は平文でなければならない
5 タグ \method は \usage 中でのみ適正である
5 タグ \dontrun は \examples 中でのみ適正である
5 タグ *tag name* は *block name* ブロック中では不正である
5 \section のタイトルは空でない平文でなければならない
5 \title の内容は平文でなければならない
3 空のセクション *tag name*
-1 エンコーディングが宣言されていない非ASCIIの内容
-1 \enc の二番目の部分に非ASCIIの内容
-3 タグ ... はコードブロック中は適正でない
-3 見かけ上エンコーディングが宣言されていない非ASCIIの内容
-3 \enc の二番目の部分に見かけ上非ASCIIの内容
-3 不必要な括弧が ... にある
-3 \method はコードブロックの外では不正である

そして \method が \S3method や \S4method に置き換えられたバリエーション.

prepare_Rd と checkRd は共にからのセクションに対するテストを持つことを注意する：checkRd 中のものはより厳格である(本質的に何も出力されないもの).

Value

これは R のエラーを通じて失敗するかもしれないが，さもなければクラス "checkRd" のオブジェクトとして返されるような警告が集められる．このクラスはユニークなメッセージをプリントするだけの print メソッドと，より深刻なメッセージだけを選択するのに使うことが出来る minlevel 引数を持つ．(これは R CMD check 中で -1 と設定される．)

可能な致命的なエラーは構文解析からのもの (たとえば，存在しないファイル，閉じられていない引用符，エンコーディングが指定されていない非ASCII入力)，又は prepare_Rd からのもの (多重の \Rdversion 宣言，不正な \encoding 又は \docType 又は \name セクション，そして欠損するか重複する \name 又は \title セクション)がある.

Author(s)

Duncan Murdoch, Brian Ripley

See Also

parse_Rd, Rd2HTML.

---

checkRdaFiles 保存イメージの詳細の報告またはそれらの再セーブ

Description

これは save により生成されたファイルの各々に対して，そのサイズ， ASCII か XDR バイナリ書式でセーブされたか，そして圧縮されているか (そしてもしそうならどのような形式でか)を報告する．

普通そうしたファイルは拡張子 '.rda' や '.RData' を持つことに関数の名前が由来する．

Usage

```
checkRdaFiles(paths)
resaveRdaFiles(paths, compress = c("auto", "gzip", "bzip2", "xz"),
               compression_level)
```

Arguments

paths	save ファイルへのパスの文字列ベクトル．もしこれが単一のディレクトリを指定すれば，それはそのディレクトリ中の全ての '.rda' そして '.RData' ファイルを参照する．
compress, compression_level	圧縮のタイプとレベル： save を見よ． compress の値は短縮出来る．

Details

compress = "auto" は R に圧縮法を選ぶように問い合わせ， compression_level を無視する． これは "gzip", "bzip2" そして，もし "gzip" 圧縮サイズが10Kbを超えれば "xz" で，最少の圧縮サイズファイルを選択する (しかし "gzip" に 10% のバイアス)． これは遅いかもしれない．

Value

checkRdaFiles に対しては，行名 paths を持つデータフレームで，列は

size	数値：バイト単位のファイルサイズ，もしファイルが存在しなければ NA.
ASCII	論理値：save(ASCII = TRUE) に対しては真で，もし書式が R のセーブファイルでなければ NA.
compress	文字列：圧縮のタイプ． "gzip", "bzip2", "xz", "none" 又は "unknown" (これはもしそれが R のセーブファイルなら R の最近のバージョン由来であることを意味する)のどれか.
version	整数：保存のバージョン – 普通 2 であるが非常に古いファイルに対しては 1 で，他のファイルに対しては NA.

Examples

```
## Not run:
## パッケージのトップレベルのソースディレクトリから
paths <- sort(Sys.glob(c("data/*.rda", "data/*.RData")))
(res <- checkRdaFiles(paths))
## より注意が必要な幾つかをピックアップ
bad <- is.na(res$ASCII) | res$ASCII | (res$size > 1e4 & res$compress == "none")
res[bad, ]

## End(Not run)
```

---

checkTnF *R* パッケージやコード中の *T/F* のチェック

Description

指定された R パッケージやコードファイルに T や F があるかどうかをチェックし，これらを含む表現式を集める． R では T と F は既定で TRUE と FALSE に設定される単なる変数であるが，予約語ではなくユーザが書き換えることが出来るため有用である．従って，論理値には常に TRUE と FALSE を使うべきである．

Usage

```
checkTnF(package, dir, file, lib.loc = NULL)
```

Arguments

package	インストール済みのパッケージの名前を与える文字列．もし与えられると，インストールされている R コードとパッケージのドキュメントファイル中の例がチェックされる．イメージファイルとしてインストールされている R コードはチェックできない．
dir	パッケージのルートソースディレクトリへのパスを指定する文字列．これは副ディレクトリ 'R' (R コード用)を含むべきであり，また 'man' (ドキュメント用)を含むべきである． package が与えられない時だけ使われる．もし使われると，ドキュメントファイル中の R コードファイルがチェックされる．
file	チェックされるべき R コードを含むファイル名． package も dir も与えられない時使われる．
lib.loc	R ライブラリのディレクトリ名の文字列ベクトル，又は NULL．既定値の NULL は現在知られている全てのライブラリに対応する．指定されたライブラリツリーは package を探すために使われる．

Value

クラス "checkTnF" のオブジェクトで，例えば T や F の存在が見つかった各ファイルに対するそれらの生起を含む表現式を含むリストである．リストの名前は対応するファイル名である．

そうしたオブジェクトに含まれる情報を見栄え良く表示する print メソッドがある．

checkVignettes　　パッケージのビニエットをチェックする

Description

パッケージの全ての Sweave ファイルを Sweave と/又は Stangle をそれらに実行することでチェックする． tangle ステップ後に見つかった全ての R のソースコードファイルは全てのコードがエラー無しに実行できるかどうかチェックするために source 関数を適用される．

Usage

```
checkVignettes(package, dir, lib.loc = NULL,
               tangle = TRUE, weave = TRUE, latex = FALSE,
               workdir = c("tmp", "src", "cur"),
               keepfiles = FALSE)
```

Arguments

package	インストールされたパッケージの名前である文字列．もし与えられると Sweave ファイルが副ディレクトリ 'doc' 中で探される．
dir	パッケージのルートソースディレクトリへのパスを指定する文字列．この副ディレクトリ 'inst/doc' 中で Sweave ファイルが探される．
lib.loc	R ライブラリのディレクトリ名の文字列，または NULL．既定値の NULL は現在知られている全てのライブラリに対応する．指定されたライブラリツリーは package を探すのに使われる．
tangle	取り出されたコードに対して tangle と source を実行するか?
weave	weave を実行するか?
latex	論理値：もし weave と latex が TRUE でビニエットのディレクトリに 'Makefile' がなければ， pdflatex を通じて weave されたファイルを実行する．
workdir	ビニエットをチェック中に作業ディレクトリとして使われるディレクトリ．もし "tmp" ならば一時ディレクトリが作られ，これが既定になる．もし "src" ならばビニエット自体を含むディレクトリが使われ，もし "cur" ならば R の現在の作業ディレクトリが使われる．
keepfiles	一時的ディレクトリ中のファイルを削除するか? workdir != "tmp" の時はこのオプションは無視される．

Details

'vignette' はパッケージの 'inst/doc' ディレクトリにある拡張子 '.Rnw' (好ましい)，'.Snw'，'.Rtex' 又は '.Stex' を持つファイルである (そして小文字バージョンもまた受け入れられる)．

もし tangle が真ならば，この関数は各ビニエットから(一つ又はそれ以上の) R のコードファイルを作り出すために Stangle を実行し，それから各コードファイルを順に source する．

もし weave が真ならば，ビニエットは Sweave を通じて実行され，各ビニエットに対して '.tex' ファイルを生成する．もし latex が又真ならば， texi2pdf が先のステップでエラーを与えなかったビニエットからの '.tex' ファイルに対して実行される．

Value

クラス "checkVignettes" のオブジェクトで， tangle, source, weave そして latex ステップで見つかったエラーメッセージを持つリストである．そうしたオブジェクト中に含まれる情報を表示する print メソッドがある．

---

check_packages_in_dir　ソースパッケージとそれらの従属性をチェックする

Description

与えられたディレクトリ中のソースパッケージをチェックする．オプションで逆向きの従属性も調べる．

Usage

```
check_packages_in_dir(dir,
                      check_args = character(),
                      check_args_db = list(),
                      reverse = NULL,
                      check_env = character(),
                      xvfb = FALSE,
                      Ncpus = getOption("Ncpus", 1L),
                      clean = TRUE,
                      ...)

summarize_check_packages_in_dir_results(dir, all = TRUE,
                                        full = FALSE)
summarize_check_packages_in_dir_timings(dir, all = FALSE,
                                        full = FALSE)
summarize_check_packages_in_dir_depends(dir, all = FALSE,
                                        which = c("Depends",
                                                  "Imports",
                                                  "LinkingTo"))

check_packages_in_dir_changes(dir, old,
                              outputs = FALSE, sources = FALSE)
```

Arguments

dir	チェックされるソース '.tar.gz' ファイルのあるディレクトリへのパスを与える文字列.
check_args	R CMD check に渡される引数の文字列ベクトル，またはそれぞれパッケージのチェックと逆向きの従属性をチェックするために使われるそうした文字列ベクトルの長さ2のリスト.
check_args_db	R CMD check に渡される引数の文字列ベクトルの名前付きリストで，名前はそれぞれのパッケージ名に対応する.
reverse	"repos", "which" または "recursive" に部分的にマッチする名前のリストで，逆向きの従属性の位置に対するレポジトリ (既定：getOption("repos"))と，逆向き従属性のタイプ (既定： "Depends", "Imports", "LinkingTo")

	package_dependencies に対するように省略形 "most" と "all" を持つ)を与える，そして逆向きの従属性の逆向きの従属性のチェック等も行うかを指示する (既定：FALSE)，または NULL (既定)で，その時は逆向きの従属性はチェックされない.
check_env	チェックに対する環境変数を設定する name=value の形式の文字列の文字列ベクトル，またはそれぞれパッケージと逆向き従属性のチェックに対して使われるそうした文字列ベクトルの長さ2のリスト.
xvfb	チェックを仮想的フレームバッファー X サーバーの内部で実行するかどうかを指示する論理値(Unix のみ)，またはそれを行うための Xvfb オプションの文字列ベクトル.
Ncpus	並列インストールとチェックとのために使われる並列プロセスの数.
clean	ダウンロードされた逆向き依存ソースを削除するかどうかを指示する論理値.
...	現在使われない.
all	チェックされた逆向き依存性も要約するかどうかを指示する論理値.
full	OK では無かったチェックに対する詳細も与えるか，またはチェック例のタイミングを要約する(もし利用可能なら)かどうかを指示する論理値.
which	package_dependencies を見よ.
old	前の check_packages_in_dir 実行のディレクトリへのパスを与える文字列.
outputs	実行されたチェックの出力中の変更を解析すべきかどうか，または(既定)チェックの状態だけかを指示する論理値.
sources	チェックされたソースファイル中の変更も調査すべきかどうかを指示する論理値 (既定値：FALSE).

Details

check_packages_in_dir は与えられたディレクトリ dir 中のソースパッケージファイル '.tar.gz' を reverse で制御されるそれらの逆向き従属性に沿って便利にチェックすることを可能にする.

reverse の "which" 成分はまたリストでもよく，その場合逆向きの従属性がリストの各成分そして対応する reverse の成分 "recursive" (これは必要に応じてリサイクルされる)に対して得られる.

もし必要なら，同様にチェックされる逆向きに従属するソースファイル '.tar.gz' dir にダウンロードされる(そしてもし clean が真なら最後に削除される). 次にチェックに対して必要な(追加の)全てのパッケージが dir の 'Library' 副ディレクトリにインストールされる. それから全ての '.tar.gz' ファイルが与えられた引数と環境変数を使いチェックされ，ファイルに対する出力とメッセージが dir の副ディレクトリ 'Outputs' 中のファイルに置かれる. 逆向き依存性のチェック結果の '*.Rcheck' ディレクトリはそれらの基礎名に接頭辞 'rdepends_' を付けて改名される.

結果とタイミングはそれぞれ summarize_check_packages_in_dir_results と summarize_check_packages_in_dir_timings を使って簡便に要約することが出来る.

もし Ncpus が1より大きければインストールとチェックは並列で行える：これは Unix では mclapply， Windows では parLapply を使う.

check_packages_in_dir はクラス "check_packages_in_dir" を継承するオブジェクトを返し，これは print と summaryメソッドを持つ.

check_packages_in_dir_changes はソース(の一部)を変更する影響を解析することを可能にする． dir と old を対応するチェック結果を持つそれぞれ新旧のディレクトリへのパスとすると，チェック結果の変更はもしあれば簡便に解析出来，オプション outputs と sources で制御できる．

Note

この機能は依然実験的である：インタフェイスは将来変更されるかもしれない．

Examples

```
## Not run:
## dir 中のパッケージを逆向きの従属性無しにチェック：
check_packages_in_dir(dir)
## dir 中のパッケージと逆向きの従属性を既定値を使いチェック
## (getOption("repos") 中の全てのディレクトリ，全ての"強い"
## 逆向き従属性，再帰的な逆向き従属性は無し)：
check_packages_in_dir(dir, reverse = list())
## dir 中のパッケージと CRAN からの逆向きの従属性を
## 全ての"強い"逆向き従属性と逆向きの推奨を使いチェック：
check_packages_in_dir(dir,
                      reverse = list(repos = getOption("repos")["CRAN"],
                                     which = "most"))
## dir 中のパッケージと CRAN からの逆向きの従属性を
## '--as-cran' を前者に対して使うが後者に対しては使わずチェック：
check_packages_in_dir(dir,
                      check_args = c("--as-cran", ""),
                      reverse = list(repos = getOption("repos")["CRAN"]))

## End(Not run)
```

---

codoc コードとドキュメントの一貫性のチェック

Description

パッケージ中の R オブジェクトの実際とドキュメント化された '構造'の間の非一貫性を見つける． codoc は名前と，オプションでまた対応する位置と関数引数の既定値を比較する． codocClasses と codocData はそれぞれ S4 クラスのスロット名とデータセットの変数名を比較する．

Usage

```
codoc(package, dir, lib.loc = NULL,
      use.values = NULL, verbose = getOption("verbose"))
codocClasses(package, lib.loc = NULL)
codocData(package, lib.loc = NULL)
```

Arguments

package	インストールされたパッケージの名前の文字列．

dir — パッケージのルートソースディレクトリへのパスを指定する文字列．これは R のドキュメントのソース(in Rd format)を持つ副ディレクトリ 'man' と R コードを持つ 'R' を含まなければならない． package が与えられない時だけ使われる．

lib.loc — R ライブラリのディレクトリ名の文字列ベクトルか NULL．既定値の NULL は現在知られている全てのライブラリに対応する．指定されたライブラリツリーは package を探すために使われる．

use.values — もし FALSE ならばコードとドキュメントを比較する際に関数の既定値を使わない．さもなければ，もし TRUE なら全ての既定値を比較し，さもなければ用法中に文章化されているものだけを比較する．

verbose — 論理値．もし TRUE なら，追加の診断がプリントされる．

Details

codoc の目的は関数オブジェクトのドキュメント化された用法と R コードで定義されたそれらの形式的引数とが一致するかどうかをチェックすることである．特に総称的関数のメソッドはしばしばメソッドではなく総称的関数の名前を用いているので，これは常に簡単なわけではない．

次のようなアルゴリズムが使われる．もしインストール済みのパッケージが使われると，もしかすると既にロードされているパッケージ(それが **base** パッケージでない限り)のバージョンを切り離した後に，それをロードする．さもなければもしソースが使われるとパッケージの R コードファイルが集められ新しい環境中で読み込まれる．それから Rd ファイルの用法セクションが抜き出されドキュメント化された形式引数を与えるために '可能な限り'構文解析される．コード環境中のインタプリタ関数に対しては，引数 use.values の値に応じて形式的引数がコードとドキュメント間で比較される．もしあれば梗概セクションが使われる；それらの存在はもし verbose が真ならば報告される．

もしパッケージが名前空間を持てば，移出オブジェクトと未移出オブジェクトの双方が登録された S3 メソッドと共にチェックされる． (ありそうもない食い違いの場合は順序はパッケージ中の移出オブジェクト，登録された S3 メソッド，そして名前空間中のオブジェクトであり，そして最初に見つかったものがチェックされる．)

現在 R のドキュメント書式は基本的なクラスの'構造'とデータセット (関数の梗概に対する用法セクションに似た) 高水準のマークアップを持たない． prompt で作られた'テンプレート'を適当に編集して得られたドキュメントオブジェクト中のデータフレームに対する変数名はドキュメントオブジェクトが単一のデータフレーム (つまり唯一つの別名を持つ)ならば codocData により認識されそして使われる．

codocClasses は同様に promptClass により作られるシェルを編集して得られるドキュメントオブジェクト中のクラスに対するスロット名を処理する．

適当な *pkgname* に対する '*pkgname*-defunct.Rd' という名前のヘルプファイルはより大雑把にチェックされる，それらはドキュメント化され無い引数を持つかもしれないからである．

Value

codoc はクラス "codoc" のオブジェクトを返す．現在これはリストであり，非一貫性が見つかったパッケージ中の各 Rd オブジェクトに対して，ミスマッチのリスト要素を含む (これは更に要素 code と docs を持つリストで，関数コードとドキュメントの用法から得られた対応する引数を与える).

codocClasses と codocData はそれぞれクラス "codocClasses" と "codocData" を返し，クラス "codoc" に類似した構造を持つ．

そうしたオブジェクトに含まれる情報を見栄え良く表示する print メソッドがある．

Note

use.values に対する既定値は R バージョン 1.9.0 とそれ以降で FALSE から NULL へ変更されている.

See Also

undoc, QC

---

compactPDF　　コンパクトな *PDF* ファイル

Description

PDF ファイル(特にビニエット)をよりコンパクトに再セーブする.
R CMD build --compact-vignettes に対するサポート関数.

Usage

```
compactPDF(paths,
           qpdf = Sys.which(Sys.getenv("R_QPDF", "qpdf")),
           gs_cmd = Sys.getenv("R_GSCMD", ""),
           gs_quality = Sys.getenv("GS_QUALITY", "none"),
           gs_extras = character())

## S3 method for class 'compactPDF'
format(x, ratio = 0.9, diff = 1e4, ...)
```

Arguments

paths	PDF ファイルへのパスの文字列ベクトル，またはそのディレクトリ中の全ての '.pdf' ファイルが使われるのならディレクトリの名前を与える長さ1の文字列ベクトル.
qpdf	qpdf 命令へのパスを与える文字列. もし空ならば， qpdf は使われない.
gs_cmd	もしそれが使われるのならば GhostScript の実行ファイルへのパスを与える文字列. Windows ではこれは 'gswin32c.exe' または 'gswin64c.exe' へのパス. もし "" (既定)ならば必要な場合は GhostScript へのプラットホーム固有のパスを見つけようと試みる.
gs_quality	必要な品質を指示する文字列：オプションは "none" (従って GhostScript は使われない)， "printer" (300dpi), "ebook" (150dpi) そして "screen" (72dpi). 短縮化出来る.
gs_extras	GhostScript に渡される追加のオプションのオプション文字列.
x	クラス "compactPDF" のオブジェクト.
ratio, diff	報告に対する限界：ファイルはそのサイズが因子 ratio と diff バイトの双方で縮小した時だけ報告される.
...	他のメソッドに/から渡される追加引数.

Details

これは既定では http://qpdf.sourceforge.net/ (Windows バイナリを含む) から入手できCRAN の R の OS X ディストリビューションに含まれる qpdf を使用する．もし gs_cmd が空でなく gs_quality != "none" ならば，先ず GhostScript が使われ，それから利用できれば qpdf が使われる．もし gs_quality != "none" で gs_cmd が "" ならば， GhostScript 実行ファイルを見つけようと試みられる．

qpdf と/または gs_cmd は見つかった全ての PDF ファイルに対して実行され，サイズで少なくとも 10% と 10Kb 減少するものだけが置き換えられる．

qpdf を使う戦略は(損失なしに) PDF ストリームとオブジェクトの双方を圧縮するというものである． GhostScript はストリームとそれ以上を圧縮し (ダウンサンプリングと埋め込みイメージの圧縮を含む)そして結果としてかなり遅く品質を失うかもしれない(しかしまたよりかなり小さな PDF ファイルを作成できる)．しかしながら，品位 "ebook" はスクリーンで眺めたりレーザープリンターでプリントするには完全に十分である．

PDF ファイルが変更される時はそれらはバージョン 1.5 の PDF ファイルになる：これらは 2003 年のバージョン6から Acrobat Reader でサポートされているので，問題を惹き起こすことはまずありそうもない．

ストリーム圧縮は最もしばしば大きな利得を持つものである．殆どの PDF ドキュメントはオブジェクトの圧縮を伴って生成されるが，しかしこれは MiKTeX の pdflatex に対しては既定ではないように見える．ある種の PDF ファイル (そして特にパッケージのビニエット)に対しては，GhostScript の使用は埋め込みイメージ(しばしばスクリーンショット)が占めるスペースを劇的に減らすことが出来る．

GhostScript と qpdf の双方が選択された時 (gs_quality != "none" で両方の実行ファイルが見つかった場合)，それらはその順序で実行されサイズの縮小は達成できた総圧縮に適用される．

Value

クラス c("compactPDF", "data.frame") のオブジェクト．これは変更されたファイルに対するバイト単位のサイズである二つの列を持つ．

メソッド format と print がある：後者は ... を書式メソッドに渡すので， ratio と diff 引数を受け入れる．

Note

使われている外部ツールは将来のリリースで変更される可能性がある．

GhostScript のバージョン 9.06 とそれ以降は CRAN パッケージのビニエットに関してバージョン 9.05 よりも数倍良い圧縮比を持つ．

See Also

resaveRdaFiles.

Adobe Acrobat (Reader では無い)を含み，PDF ファイルをコンパクト化する多くの他の(そしてある場合にはより効率的な)ツールが利用できる．マニュアル 'Writing R Extensions' を見よ．

delimMatch 区切り文字のパターンマッチ

Description

文字列ベクトル中の区切り文字で囲まれた副文字列を適正な入れ子でマッチする.

Usage

```
delimMatch(x, delim = c("{", "}"), syntax = "Rd")
```

Arguments

x	文字列ベクトル.
delim	長さ2の文字列ベクトルで開始と終了の区切り文字を与える. 将来は任意の正規表現が許されるかもしれない.
syntax	現在は常に文字列 "Rd" で Rd シンタックスを指示する(つまり '%' は行末までのコメントで, '\' はエスケープ). 将来のバージョンは恐らく柔軟にコメント, エスケープ, そして引用符を指定することを許す'シンタックス表'を使った他のシンタックスを理解するかもしれない.

Value

x と同じ長さの整数ベクトルで最初のマッチの位置 (文字数単位), またはもしなければ -1 を与え, マッチしたテキスト (もしくはマッチするものがなければ -1)の長さを与える属性 "match.length" を持つ.

See Also

'単純'なパターンマッチングには regexpr.

Examples

```
x <- c("\\value{foo}", "function(bar)")
delimMatch(x)
delimMatch(x, c("(", ")"))
```

dependsOnPkgs 逆向きの従属性を見つける

Description

パッケージの'逆向き'従属性を見つける, つまりこのパッケージに依存するもの, そして (オプションで)再帰的に探索する.

Usage

```
dependsOnPkgs(pkgs,
              dependencies = c("Depends", "Imports", "LinkingTo"),
              recursive = TRUE, lib.loc = NULL,
              installed =
              utils::installed.packages(lib.loc, fields = "Enhances"))
```

Arguments

pkgs	パッケージ名の文字列ベクトル.
dependencies	従属性のタイプをリストする文字列ベクトルで c("Depends", "Imports", "LinkingTo", "Suggests", "Enhances") の部分集合. 文字列 "all" はこのベクトルの短縮形で, そして "most" は現在 "Enhances" を除くもの.
recursive	論理値：逆向き従属性の逆向き従属性(等)を含めるべきか?
lib.loc	R のライブラリツリーの文字列, または全ての既知のツリーに対する NULL (.libPaths を見よ).
installed	呼び出し installed.packages の結果.

Value

パッケージ名の文字列で, pkgs からのものは含まない.

Examples

```
## バニラ R インストールにはほとんど従属性がない：
## lattice はインストールされないかもしれない
dependsOnPkgs("lattice")
```

---

encoded_text_to_latex 非アスキー文字を *LaTeX* のエスケープシーケンスに翻訳

Description

非アスキー文字を LaTeX のエスケープシーケンスに翻訳する.

Usage

```
encoded_text_to_latex(x,
                      encoding = c("latin1", "latin2", "latin9",
                                   "UTF-8", "utf8"))
```

Arguments

x	文字列ベクトル.
encoding	仮定されるエンコーディング. "latin9" は公式には ISO-8859-15 または Latin-9 であるが, LaTeX の inputenc パッケージでは latin9 として知られている.

Details

x 中の非アスキー文字が適当な LaTeX のエスケープシーケンスに置き換えられる，もしくは適当なシーケンスがなければ '?' に置き換えられる．

たとえ適当なシーケンスがあっても，それは使用中のフォントでサポートされないかもしれない．ハイフンは '\-' にマップされる．

Value

x と同じ長さの文字列ベクトル．

See Also

iconv

Examples

```
x <- "fa\xE7ile"
encoded_text_to_latex(x, "latin1")
## Not run:
## 8ビット文字の上半分を示すテキストを作る
x <- rawToChar(as.raw(160:255), multiple = TRUE)
(x <- matrix(x, ncol = 16, byrow = TRUE))
xx <- x
xx[] <- encoded_text_to_latex(x, "latin1") # または latin2 や latin9
xx <- apply(xx, 1, paste, collapse = "&")
con <- file("test-encoding.tex", "w")
header <- c(
"\\documentclass{article}",
"\\usepackage[T1]{fontenc}",
"\\usepackage{Rd}",
"\\begin{document}",
"\\HeaderA{test}{}{test}",
"\\begin{Details}\relax",
"\\Tabular{cccccccccccccccc}{")
trailer <- c("}", "\\end{Details}", "\\end{document}")
writeLines(header, con)
writeLines(paste0(xx, "\\"), con)
writeLines(trailer, con)
close(con)
## そして UTF_8 文字を幾つか
x <- intToUtf8(as.integer(
    c(160:383,0x0192,0x02C6,0x02C7,0x02CA,0x02D8,
      0x02D9, 0x02DD, 0x200C, 0x2018, 0x2019, 0x201C,
      0x201D, 0x2020, 0x2022, 0x2026, 0x20AC)),
               multiple = TRUE)
x <- matrix(x, ncol = 16, byrow = TRUE)
xx <- x
xx[] <- encoded_text_to_latex(x, "UTF-8")
xx <- apply(xx, 1, paste, collapse = "&")
con <- file("test-utf8.tex", "w")
writeLines(header, con)
writeLines(paste(xx, "\\", sep = ""), con)
writeLines(trailer, con)
close(con)

## End(Not run)
```

`fileutils` ファイルユーティリティ

Description

ファイルのりストとファイルパス操作のためのユーティリティ.

Usage

```
file_ext(x)
file_path_as_absolute(x)
file_path_sans_ext(x, compression = FALSE)

list_files_with_exts(dir, exts, all.files = FALSE,
                     full.names = TRUE)
list_files_with_type(dir, type, all.files = FALSE,
                     full.names = TRUE, OS_subdirs = .OStype())
```

Arguments

`x`	ファイルパスを与える文字列ベクトル.
`compression`	論理値：圧縮拡張子 '.gz', '.bz2' または '.xz' は最初に取り除くべきか?
`dir`	ディレクトリへのパス名の文字列.
`exts`	可能なファイル拡張子の文字列ベクトル(先頭のドットを除く).
`all.files`	論理値. もし `FALSE` (既定)ならば, 可視ファイルだけが考慮される ; もし `TRUE` ならば全てのファイルが使われる.
`full.names`	見つかったファイルの完全なパスを返す(既定値)べきかどうか, またはファイル名だけかを指示する論理値.
`type`	拡張子で特徴付けられるリストされるファイルの 'タイプ'を与える文字列. 現在可能な値は `"code"` (R コード), `"data"` (データセット), `"demo"` (デモ), `"docs"` (R ドキュメント) そして `"vignette"` (ビニエット).
`OS_subdirs`	R コードとドキュメントファイルのリストで可能性として含められる OS 固有の副ディレクトリの名前の文字列. 既定では環境変数 `R_OSTYPE` の値か, もしくはもしこれが空なら `.Platform$OS.type` の値が使われる.

Details

`file_ext` はファイル(名)拡張子を返す(先頭のドットを除く). (純粋なアルファニューメリックな拡張子だけが認識される.)

`file_path_as_absolute` は相対パスがあれば絶対パスに変更し, 必要ならチルダ展開を行う. これは `normalizePath` に対するラッパである. `x` は単一の既存のパスでなければならない.

`file_path_sans_ext` は拡張子 (と先頭のドット)無しのファイルパスを返す. (純粋なアルファニューメリックな拡張子だけが認識される.)

`list_files_with_exts` はディレクトリ `dir` 中の拡張子が `exts` の要素の一つにマッチするファイルのパスか名前を返す. 既定では完全なパスが返され, 可視ファイルだけが使われることを注意する.

list_files_with_type は，R が認識する拡張子で決まるような， dir 中の与えられた‘タイプ’のファイルのパスを返す． R コードとドキュメントファイルをリストするときは，もしあれば OS 固有の副ディレクトリ中のファイルが OS_subdirs の値に応じて含められる．既定では完全なパスが返され，可視ファイルだけが使われることを注意する．

See Also

file.path, file.info, list.files

Examples

```
dir <- file.path(R.home(), "library", "stats")
list_files_with_exts(file.path(dir, "demo"), "R")
list_files_with_type(file.path(dir, "demo"), "demo") # 同じ
file_path_sans_ext(list.files(file.path(R.home("modules"))))
```

---

find_gs_cmd *GhostScript* の実行ファイルを見つける

Description

GhostScript の実行ファイルをプラットホームを跨ぐような仕方で見つける．

Usage

```
find_gs_cmd(gs_cmd = "")
```

Arguments

gs_cmd	名前，GhostScript 実行ファイルの完全か部分的なパス．

Details

詳細はプラットホームにより異なる．

Unix 風システムでは GhostScript の実行ファイルは普通 gs と呼ばれる．命令の名前(そしてもしかするとパス) は先ず gs_cmd から，そして環境変数 R_GSCMD から取られ，既定値は gs である．これはそれからシステムパス上で探され，そしてもしマッチするものが見つかれば返される．

Windows では，命令の名前は引数 gs_cmd そしてそれから環境変数 R_GSCMD と GSC から取られる．もしこれらのどれも適当な命令名を与えなければ， gswin64c と gswin32c が順番に試みられる．全ての場合で命令はシステム PATH 上で探される．

Windows (そしてある種の他の OS)上では Postscript/PDF ファイルを表示するのとそれらを操作するものという別個の GhostScript 実行ファイルが存在することを注意する：この関数は後者を探す．

Value

もし見つかれば GhostScript 実行ファイルへの完全なパスを与える文字列で，さもなければ空の文字列．

Examples

```
## Not run:
## Solaris システムがパス上に GhostScript 9.00 を持ち
## /opt/csw/bin に 9.07 を持つと仮定する.
## すると次のように設定できるかもしれない
Sys.setenv(R_GSCMD = "/opt/csw/bin/gs")

## End(Not run)
```

---

getDepList 従属性情報を検索する関数

Description

これらの関数はパッケージの従属性情報や従属性行列に対する DependsList オブジェクトの情報を検索する.

従属性行列をあたえた時, getDepList() はそのパッケージに対する DependsList オブジェクトを作る. このオブジェクトは行列に対する従属性, 何がインストールされているか, どのような未解決の従属性がオンラインで見つかったか, そして任意の R 従属性を含む.

pkgDepends() は便宜上パッケージ名を取り, 従属性行列を構築し, それから対応する getDepList() を返す.

Usage

```
getDepList(depMtrx, instPkgs, recursive = TRUE, local = TRUE,
           reduce = TRUE, lib.loc = NULL)

pkgDepends(pkg, recursive = TRUE, local = TRUE, reduce = TRUE,
           lib.loc = NULL)
```

Arguments

depMtrx	package.dependencies() が作るような従属性行列.
pkg	パッケージ名.
instPkgs	package.dependencies が作るようなローカルシステム上の全てのインストールされたパッケージを指定する行列.
recursive	間接的従属性を含めないかどうか.
local	ローカルのみ創作するかどうか.
reduce	従属性のセットを最少の値に簡略化するかどうか.
lib.loc	インストールされたパッケージを探す時どのライブラリを使うか. NULL は現在の .libPaths() 中の全てのライブラリディレクトリを指示する. lib.loc は getDepList() 中とそこで廃止予定とされたものでは使われないことを注意する.

Details

便利関数 pkgDepends(pkg) は installed.packages を問い合わせ従属性行列を作り，この情報を使い getDepList を呼び出し，そして結果を返す．

もし recursive が TRUE ならば，従属していると指示された任意のパッケージは次にその従属性を含む(等々)，これらは間接的従属性として知られている．もし recursive が FALSE ならば，パッケージが直接指示する従属性が使われる．

もし local が TRUE ならば，システムは未解決の従属性を見つけるのにユーザがローカルにインストールしているものだけを探す．

もし reduce が TRUE ならば，システムは最少の従属性のセットが指定されるように DependsList オブジェクト中の欄を縮小する (例えばもし 'foo, foo (>= 1.0.0), foo (>= 1.3.0)' ならば，'foo (>= 1.3.0)' が返される)．

Value

S3 クラス "DependsList" のオブジェクトで，これは次の成分を含むリストである

Depends	このパッケージに対する従属性のベクトル．
Installed	現在インストール済みのパッケージが満足する従属性のベクトル．
Found	Installed 中には無いがオンラインで見つかった従属性を表現するリスト．このリストはパッケージが見つかったレポジトリに対する URL である要素名を持ち要素自体はそれぞれのレポジトリで見つかったパッケージ名のベクトルである．もし local = TRUE ならば Found 要素は常に空である．
NotFound	...
R	任意の R のバージョンの従属性．

Author(s)

Jeff Gentry

See Also

installFoundDepends

Examples

```
pkgDepends("tools", local = FALSE)
```

---

getVignetteInfo　　インストールされたビニエットの情報を得る

Description

この関数はインストールされたビニエットの情報を得る．

Usage

```
getVignetteInfo(package = NULL, lib.loc = NULL, all = TRUE)
```

Arguments

package	どのパッケージを調べるか，または NULL なら全てのパッケージ.
lib.loc	どのライブラリを調べるか.
all	全てのインストールされたパッケージを検索するか，それとも単に付加されたパッケージだけか.

Value

次の列を持つ行列

Package	パッケージ名
Dir	パッケージがインストールされているディレクトリ
Topic	ビニエットの名前
File	ビニエットのソースの基本ファイル名
Title	ビニエットのタイトル
R	ビニエットからの tangle された R ソース
PDF	表示用の PDF または HTML ファイル

Note

結果の最後の列は歴史的理由から PDF という名前を持つが， PDF や HTML ドキュメントのファイル名を含むかもしれない.

See Also

pkgVignettes はインストールされていないパッケージに対して使える同様の関数である.

Examples

```
getVignetteInfo("grid")
```

HTMLheader *R* ヘルプに対する標準的 *HTML* ヘッダーを生成する

Description

この関数は R のヘルプページで使われる標準的な HTML ヘッダーを生成する.

Usage

```
HTMLheader(title = "R", logo = TRUE, up = NULL,
           top = file.path(Rhome, "doc/html/index.html"),
           Rhome = "",
           css = file.path(Rhome, "doc/html/R.css"),
           headerTitle = paste("R:", title),
           outputEncoding = "UTF-8")
```

Arguments

title	HTML ヘッダー中で表示され使われるタイトル．任意の HTML エスケープが既になされているべきである．
logo	タイトルの後に R ロゴを表示すべきか．
up	“up” ボタンでどのページ(もしあれば)にリンクするか．
top	“top” ボタンでどのページ(もしあれば)にリンクするか．
Rhome	R のホームディレクトリへの相対パス．‘詳細’節を見よ．
css	階層的スタイルシートに対する相対的 URL．
headerTitle	ヘッダー中で使われるタイトル．
outputEncoding	ページ全体に対して宣言されたエンコーディング．

Details

リンク up と top は現在のページに関して相対的である． Rhome パスは既定では動的なヘルプと共に動作する；静的なヘルプに対してはそれに対する相対的なパス(例えば‘../..’)を使うべきである．

Value

HTML ヘッダー行を含む文字列ベクトルで R のヘルプシステムを開始するのに使える．

Examples

```
cat(HTMLheader("This is a sample header"), sep="\n")
```

HTMLlinks　　パッケージドキュメントからの *HTML* リンクを集める

Description

他のパッケージのインストールされた HTML ドキュメントへの URL に対する相対ファイルパスを計算する．

Usage

```
findHTMLlinks(pkgDir = "", lib.loc = NULL, level = 0:2)
```

Arguments

pkgDir	インストール済みのパッケージのトップディレクトリ．既定はどのパッケージも指示しない．
lib.loc	スキャンする R ライブラリツリーの位置を記述する文字列：既定は .libPaths() を指示する．
level	どのレベルをインクルードするか．

Details

findHTMLlinks は一つのヘルプページから別へのリンクを解決しようと試みる．これは次の減少順の優先度を使う

- pkgDir 中のパッケージ：これはそのパッケージ(レベル 0)に対する HTML ヘルプの変換時に使われる．
- 基本と推奨パッケージ(レベル 1).
- lib.loc で指定されるライブラリツリー中でツリーの順序そして一つのライブラリツリー中ではアルファベット順で見つかる他のパッケージ (レベル 2).

Value

ファイルパスの名前付き文字列ベクトルで，インストール済みのパッケージの 'html' ディレクトリに対して相対的．従ってこれらは形式 '"../../*somepkg*/html/*sometopic*.html"' でなければならない．

Author(s)

Duncan Murdoch, Brian Ripley

---

installFoundDepends　未解決の従属性をインストールする関数

Description

この関数は pkgDependsList オブジェクトの Found 要素を取り指定されたレポジトリから全てのリストされたパッケージをインストールしようとする．

この関数は pkgDependsList オブジェクトが廃止予定なため廃止予定である．

Usage

```
installFoundDepends(depPkgList, ...)
```

Arguments

depPkgList	pkgDependsList オブジェクトからの Found 要素
...	install.packages に渡される引数

Details

この関数は入力として pkgDependsList オブジェクトからの Found リストを取る．このリストは対応するレポジトリの URL の名前の要素を持ち，要素はパッケージ名のベクトルである．各要素に対してベクトル中にリストされた全てのパッケージをインストールするために install.packages がその URL に対して呼び出される．

Author(s)

Jeff Gentry

See Also

pkgDepends, install.packages

Examples

```
## インストールするパッケージに対する一時ディレクトリを設定する
tmp <- tempfile()
dir.create(tmp)

pDL <- pkgDepends("tools", local = FALSE)
installFoundDepends(pDL$Found, destdir = tmp)
```

---

loadRdMacros　　ユーザ定義の *Rd* ヘルプシステムマクロをロードする

Description

'.Rd' ファイルからマクロをロードする，もしくはパッケージに含まれる複数の '.Rd' ファイルをロードする．

Usage

```
loadRdMacros(file, macros = TRUE)
loadPkgRdMacros(pkgdir, macros)
```

Arguments

file	マクロ定義を含む Rd 書式のファイル．
macros	TRUE または parse_Rd の macros 引数が期待する書式のマクロ定義の以前のセット．
pkgdir	ソースパッケージかインストールされたパッケージの基本ディレクトリ．

Details

この関数により構文解析されるファイルはマクロ定義だけを含むべきである；コメントや空白以外の何かが見つかれば警告が出る．

macros 引数はマクロの基礎セットの名前ファイルや以前の loadRdMacros 呼び出しの結果や同じセッションの loadPkgRdMacros で良い．これらの結果は現在の中でだけ成り立つと仮定されるべきである．

loadPkgMacros 関数は最初パッケージの 'DESCRIPTION' ファイル中の "RdMacros" 項目を探す．もし存在すれば，それは他のパッケージ名のコンマで分離されたリストを含むべきである；それらのマクロは現在のパッケージのそれらの前にロードされる．それから現在のパッケージ中で副ディレクトリ 'man/macros' または 'help/macros' 中で '.Rd' ファイルを探し，そしてそれらをロードする．

Value

これらの関数は最後に処理されたファイルからの新しく定義されたマクロの名前を持つオブジェクトを含む環境を各々返す．親環境は以前のファイルからのマクロである，等々．最初に処理されるファイルは emptyenv() をその親として持つ．

Author(s)

Duncan Murdoch

References

Rd ファイルのシンタックスについてはマニュアル 'Writing R Extensions' を見よ，または技術的議論については https://developer.r-project.org/parseRd.pdf を見よ．

See Also

parse_Rd

Examples

```
f <- tempfile()
writeLines(paste0("\\newcommand{\\logo}{\\if{html}{\\figure{Rlogo.svg}{options: width=100}",
                  "\\if{latex}{\\figure{Rlogo.pdf}{options: width=0.5in}}}"),
    f)
m <- loadRdMacros(f)
ls(m)
ls(parent.env(m))
ls(parent.env(parent.env(m)))
```

---

makevars　　ユーザとサイトのコンパイル用変数

Description

パッケージのコンパイルをカスタム化するためにユーザとサイト固有の 'Makevars' の位置を決定する．

Usage

```
makevars_user()
makevars_site()
```

Details

パッケージの保守者はこれらの関数をユーザとサイトの特定のコンパイル設定を R の機構を使わないコンパイルにも使うことに用いることが出来る (特に副ディレクトリ 'src' 中のカスタムコンパイル)，つまり R を cat(tools::makevars_user()) や cat(tools::makevars_site()) を用いて呼び出すコンフィギュレーションコードを加えたり，'-f' を用いた Make の起動に空でないこれを渡すことにより．

Value

ユーザとサイト固有の 'Makevars' ファイルへのパスを持つ文字列，またはもしそうしたファイルがなければ空文字列を返す．の位置を決定する．

See Also

マニュアル “R Installation and Administration” 中の “Customizing package compilation” 節.

Examples

```
makevars_user()
makevars_site()
```

---

make_translations_pkg 現在の変更を R ソース中にパッケージする

Description

更新した変更のパッケージを用意する R コアメンバー用のユーティリティ，

Usage

```
make_translations_pkg(srcdir, outDir = ".", append = "-1")
```

Arguments

srcdir	R のソースディレクトリ.
outDir	準備したパッケージをその中に置くディレクトリ.
append	パッケージのバージョン番号に対する接尾辞，例えば 3.0.0-1 は R 3.0.0 中では既定値.

Details

これは現在の R ソースのディストリビューション中の変更を取り出しそしてそれらを CRAN で配布し update.packages でインストール出来る **translations** と呼ばれるソースパッケージにパッケージする．これは例えば R 3.x.y で出荷された変更を現在 ‘R-patched’ 中にあるもので更新することを，管理特権を持たないユーザにも許す.

パッケージは単一の x に対するバージョン 3.x.* に制限する ‘Depends’ 欄を持つ.

---

md5sum *MD5* チェックサムを計算する

Description

一つまたはそれ以上のファイルの 32 バイトの MD5 ハッシュを計算する.

Usage

```
md5sum(files)
```

Arguments

files	文字列．その内容がハッシュされるファイルへのパス.

Details

MD5 'ハッシュ' または'チェックサム'または'メッセージダイジェスト' は32個の16進数で表現されるファイル内容の 128 ビット要約である．異なった MD5 サムのファイルは異なる：非常に稀に (そして普通誤解させる意図で) 異なったファイルが同じサムを持つことがある．

Windows では全てのファイルはバイナリモードで読み取られる (そこでの `md5sum` ユーティリティがそうする)：他の OS ではファイルは既定のモードで読み取られる (もしひとつ以上のテキストがあればテキストモードで)．

MD5 サムは R パッケージが正しく展開されその後に修正されていないかをチェックするために使われる．

Value

`files` と同じ長さの文字列ベクトルで，名前は `files` に等しい．要素は存在しないファイルや読み取れないファイルに対しては `NA` で，さもなければ16進の 32 文字の文字列．

Source

基礎にある C コードは Ulrich Drepper により書かれ `glibc` の2001年のリリースから取られた．

See Also

`checkMD5sums`

Examples

```
as.vector(md5sum(dir(R.home(), pattern = "^COPY", full.names = TRUE)))
```

---

package.dependencies　パッケージの従属性をチェック

Description

パッケージを構文解析し現在インストールされているバージョンの R (そして他のパッケージ)に対してチェックする．

Usage

```
package.dependencies(x, check = FALSE,
                     depLevel =
                     c("Depends", "Imports", "Suggests"))
```

Arguments

`x`	`available.packages` が返すようなパッケージ記述の行列．
`check`	もし `TRUE` なら，チェック結果の論理値ベクトルを返す．もし `FALSE` ならば構文解析された従属性のリストを返す．
`depLevel`	`Depends` や `Suggests` レベルの従属性を探すかどうか．短縮化出来る．

Details

現在パッケージが現在稼働中の R のバージョンに適合しているかどうかだけをチェックする．将来はパッケージ間の従属性のチェックを加えるかもしれない．

See Also

update.packages

---

package_dependencies　パッケージの従属性の階層を計算

Description

パッケージの(再帰的)従属性や逆向き従属性を見つける．

Usage

```
package_dependencies(packages = NULL, db = NULL,
        which = c("Depends", "Imports", "LinkingTo"),
recursive = FALSE, reverse = FALSE, verbose = getOption("verbose"))
```

Arguments

packages	パッケージ名の文字列ベクトル．
db	available.packages() からのような文字列ベクトル(既定の NULL ではこの呼び出しの結果) またはそれのデータフレーム版．もしくは https://cran.r-project.org/web/packages/packages.rds から得られるようなものに似たパッケージのデータベース．
which	従属性のタイプをリストする文字列ベクトルで，c("Depends", "Imports", "LinkingTo", "Suggests", "Enhances") のサブセット．文字列 "all" はこのベクトルの省略形であり，文字列 "most" は "Enhances" を除いた同じベクトルである．
recursive	論理値： (逆向き)従属性の(逆向き)従属性(等々)を含めるべきか?
reverse	論理値：もし FALSE ならば(既定)，通常の従属性が，さもなければ逆向き従属性が計算される．
verbose	出力がパッケージの探索サイクルをモニターすべきかどうかを指示する論理値．

Value

引数 packages 中の各パッケージに対して一つの要素を持つ名前付きリストで，各々はパッケージの (再帰的) (逆向き)従属性を命名する文字列ベクトルからなる．

データベースに見つからない与えられたパッケージに対しては項目 NULL が返され，従属性がなければ項目 character(0) が返される．

See Also

dependsOnPkgs そして従属性のチェックには package.dependencies.

Examples

```
## Not run:
pdb <- available.packages()
deps <- package_dependencies(packages = "MASS", pdb,
                    which = c("Depends", "Imports", "LinkingTo", "Suggests"),
                    recursive = TRUE, reverse = TRUE)
length(deps$MASS)

## End(Not run)
```

---

parseLatex　　これらの実験的な関数は *LaTeX* コードのサブセットを扱う

Description

parseLatex 関数は LaTeX のソースを構文解析し，構造化されたオブジェクトを作る；deparseLatex は過程を逆転する． latexToUtf8 関数は LaTeX オブジェクトを取り，それらを対応する UTF-8 文字に変換する幾つかの異なったマクロを処理する．

Usage

```
parseLatex(text, filename = deparse(substitute(text)),
           verbose = FALSE,
           verbatim = c("verbatim", "verbatim*",
                        "Sinput", "Soutput"))
deparseLatex(x, dropBraces = FALSE)
latexToUtf8(x)
```

Arguments

text	LaTeX のソースコードを含む文字列ベクトル．
filename	構文エラーのメッセージ中で使われるファイル名．
verbose	もし TRUE なら，デバッグエラーメッセージをプリントする．
verbatim	逐語的なテキストを保持する LaTeX 環境の名前を含む文字列ベクトル．
x	"LaTeX" オブジェクト．
dropBraces	"LaTeX" オブジェクトを表示する時不要な括弧を取り去る．

Details

構文解析器は全ての適正な LaTeX コードを認識すせず，限られた簡単な例だけである．それは引数にマクロを関連させない，これは構文解析後に各マクロの定義の知識を使いなされる必要があある．この関数の意図は完全な一般的な LaTeX ドキュメントではなく，文献参照で使われる簡単なコードを処理することである．

注釈の多いテキストは二つの形式で許される： \verb マクロ(単一の文字区切りを使い)，そしてその名前が verbatim 引数中でリストされる環境である．

Value

parseLatex() 関数はクラス "LaTeX" の再帰的なオブジェクトを返す．このオブジェクト中の項目の各々はそのシンタックス的な役割を特定する "latex_tag" 属性を持つ．

deparseLatex() 関数は単一要素の文字列ベクトルを返し，もしかすると埋め込みの新行を含むかもしれない．

latexToUtf8() 関数はそれに渡される "LaTeX" オブジェクトの修正バージョンを返す．

Author(s)

Duncan Murdoch

Examples

```
latex <- parseLatex("fa\\c{c}ile")
deparseLatex(latexToUtf8(latex))
```

---

parse_Rd　　*Rd* ファイルを構文解析する

Description

この関数は他の関数による処理のために R のドキュメント(Rd)ファイルを読み取りそれを構文解析する．

Usage

```
parse_Rd(file, srcfile = NULL, encoding = "unknown",
         verbose = FALSE, fragment = FALSE, warningCalls = TRUE,
 macros = file.path(R.home("share"), "Rd", "macros", "system.Rd"),
         permissive = FALSE)
## S3 method for class 'Rd'
print(x, deparse = FALSE, ...)
## S3 method for class 'Rd'
as.character(x, deparse = FALSE, ...)
```

Arguments

file	ファイル名かテキストモードのコネクション．現在ファイル名が最も上手く動作する．
srcfile	NULL または "srcfile" オブジェクト．‘詳細’節を見よ．
encoding	入力文字に対して仮定されるエンコーディング．
verbose	詳細な構文解析情報がプリントされるべきかどうかを指示する論理値．
fragment	ファイルが完全な Rd ファイルを表すか断片かを指示する論理値．
warningCalls	論理値：構文解析の警告は呼び出しを含めるべきか?
macros	追加のマクロをそこからロードするファイル名か環境，もしくは論理値．‘詳細’節を見よ．

permissive	認識できないマクロは警告なしでテキストとして扱われるべきことを指示する論理値.
x	クラス Rd のオブジェクト.
deparse	もし TRUE なら，結果の文字が同じオブジェクトに構文解析されるようにエスケープシーケンスを回復しようと試みる.
...	他のメソッドへ・から渡される追加引数.

Details

この関数は https://developer.r-project.org/parseRd.pdf で与えられた仕様に従って 'Rd' ファイルを構文解析する.

各構文エラーに対して警告が生成され構文解析を続行しようと試みる. 続行するためには一般にファイルのある部分を取り除く必要があるので，そうしたそうした警告は無視されるべきではない.

マークされたエンコーディングが無いファイルは既定ではネイティブなエンコーディング中にあると仮定される. 別の既定値は encoding 引数を使って設定できる. ファイル中の全てのテキストは構文解析されたオブジェクト中では UTF-8 エンコーディングに翻訳される.

R のバージョン 3.2.0 から，ユーザが定義したマクロを '\newcommand' または '\renewcommand' を使って別のファイル中に与えることができる. 環境もまた与えることが出来る：それは loadRdMacros, loadPkgRdMacros または以前の parse_Rd の呼び出しで作ることが出来る. もし論理値が与えられると. 既定の組み込みのマクロだけが使われる； FALSE は結果には "macros" 属性を返さないことを指示する.

permissive 引数は完全には Rd 書式ではないテキストの構文解析を許す. 典型的にはそれは bibentry 中におけるような， Rd 断片中で使われる LaTeX コードである. permissive = TRUE では，これは平文として渡される. parse_Rd はどれだけ多くの引数が LaTeX マクロに属するか知らないので，それはマクロの後の括弧の存在に基づいて推測する；これは絶対確かではない.

Value

parse_Rd はクラス "Rd" のオブジェクトを返す. このオブジェクトの内部書式は変更されるかもしれない. このクラスに対して定義された as.character() と print() メソッドはそれぞれ文字列ベクトルを返しそれらをプリントする.

macros = FALSE で無い限り，オブジェクトは "macros" という名前の属性を持ち，これは file 中で定義された同じセッション中の更なる parse_Rd 呼び出しに対して使うことが出来る書式で書かれたマクロを含む環境である. もしファイルに保存され異なったセッションで再ロードされると動作するかどうかは保証されない.

Author(s)

Duncan Murdoch

References

https://developer.r-project.org/parseRd.pdf

See Also

変換のために parse_Rd()の出力を利用するRd2HTML. .

pskill　　プロセスを停止する

Description

pskill はプロセスにシグナルを送り，普通それを停止する．

Usage

```
pskill(pid, signal = SIGTERM)

SIGHUP
SIGINT
SIGQUIT
SIGKILL
SIGTERM
SIGSTOP
SIGTSTP
SIGCHLD
SIGUSR1
SIGUSR2
```

Arguments

pid	整数： Sys.getpid が返すような一つまたはそれ以上のプロセス ID.
signal	整数，最もしばしばシンボリックな定数の一つ.

Details

シグナルは C99 の概念であるが，僅かな数のサポートだけが要求されている (リストされたものの内 SIGINT と SIGTERM だけ)．これらは POSIX オペレーティングシステムではより幅広く使われ (ここでリストされた全てが定義される必要がある)，またプロセスにシグナルを送る kill システムコールもサポートしており，最もしばしばそれを停止する．関数 pskill はラッパを提供する：これはゼロや負の pid を含むその引数の不正な値を黙って無視する．

Unix 風システムでの普通の用法では，Ctrl-C は SIGINT， Ctrl-\ は SIGQUIT そして Ctrl-Z は SIGTSTP を送る：これと SIGSTOP はプロセスを一時停止し SIGCONT で再開できる．

シグナルは小さな整数であるが，実際の数値は標準化されていない (そして多くが OS 間で異なる)．SIG* オブジェクトは現在のプラットホームに対して適当な整数値を含む (またはシグナルが定義されていない時は NA_INTEGER_).

Windows では SIGINT と SIGKILL だけが定義されており， pskill は常に Windows のシステムコール TerminateProcess を使う．

Value

pid, TRUE (成功に対して)または FALSE と同じ長さの論理値ベクトルで不可視.

See Also

パッケージ **parallel** は子プロセスを起動する幾つかの手段を持ち，プロセス ID を記録する．

psnice

Examples

```
## Not run:
pskill(c(237, 245), SIGKILL)

## End(Not run)
```

---

psnice　　プロセスの優先度*(niceness)*を得る，設定する．

Description

現在のプロセスまたは一つもしくは複数の他のプロセスのプロセスの‘優先度(niceness)’を得る，設定する．

Usage

```
psnice(pid = Sys.getpid(), value = NA_integer_)
```

Arguments

pid	正整数：一つまたはそれ以上のプロセスのID：既定では R のセッションプロセスのそれ．
value	設定される優先度，または問い合わせには NA．

Details

POSIX オペレーティングシステムはプロセス優先度の概念を持ち，普通 0 から 39 (または 40)で 20 が正常の優先度であり (些か混乱させるが)より大きな数値はより低い優先度を意味する．更に混乱させることに，‘niceness’ 値があり，これは優先度が数値的に 20 を超える量である(これは負になり得る)．高い niceness を持つプロセスは正常な優先度よりもより少ない CPU 時間を割り当てられる．ある種の OS では niceness が +19 のプロセスはシステムがさもなければアイドル状態の時だけ実行される．

top のような多くの OS ユーティリティは niceness ではなく優先度を報告する． niceness はユーティリティ ‘/usr/bin/renice’ により使われる： ‘/usr/bin/nice’ (そして /usr/bin/renice -n)は niceness の増分を指定する．

特権を持ったユーザ(usually普通スーパーユーザ)だけが優先度を低下出来る．

Windows は少し異なった ‘優先度クラス’の概念を持つ．アイドリングの優先度は 19, ‘正常未満’は 15, 正常は 0, ‘正常以上’は -5 そして‘実時間’は -10 にマップされる． Unix 風システムとは異なり，Windows では非特権的ユーザが優先度クラスを増加できる (しかし‘実時間’の使用は推奨できない)．

Value

直前の優先度値の整数ベクトル．何らかの理由で未知なら NA．

See Also

パッケージ **parallel** 中の様々な関数はその優先度の変更が必要かもしれない子プロセスを作る.

pskill.

QC *R* コードやドキュメントの品質管理チェック

Description

R コードとドキュメント，特に R パッケージに対して様々な品質管理 (QC)チェックを実行する関数.

Usage

```
checkDocFiles(package, dir, lib.loc = NULL)
checkDocStyle(package, dir, lib.loc = NULL)
checkReplaceFuns(package, dir, lib.loc = NULL)
checkS3methods(package, dir, lib.loc = NULL)

langElts
nonS3methods(package)
```

Arguments

package	インストール済みのパッケージ名前の文字列ベクトル.
dir	パッケージのルートソースディレクトリへのパスを指定する文字列. これはサブディレクトリ 'R' (R コード用) と R ドキュメントのソース 'man' (Rd 書式の)を含まなければならない． package が与えられない時だけ使われる.
lib.loc	R ライブラリのディレクトリ名の文字列，又は NULL. 既定値 NULL は現在既知の全てのライブラリに対応する．指定されたライブラリツリーは package を探すために使われる.

Details

checkDocFiles はパッケージ中の全ての Rd ファイルに対するチェックを行い，Rd ファイルの用法節に示された全ての引数がその引数節中にドキュメント化されているかどうかを示す．これは又引数節に重複する項目があるかどうかと引数節に与えられているが用法節には無い'過剰にドキュメント化' された引数を報告する．用法は表示されるものであるので，マッチは用法節に対して行われ，もしかすると存在するかもしれない一覧節に対してではないことを注意する.

checkDocStyle は(S3)メソッドがパッケージ中の Rd ファイルの用法節でどのように示されるかを調べる．これは S3 メソッドを指示する Rd の \method マークアップを使用せずそれらのフルネームで表示されるメソッドを報告する．初期の R はまたそれらの総称的関数と共に示されるメソッドを報告したが，これは典型的に総称的関数とそのメソッド中の主要引数のドキュメント化に対して問題を引き起こした． \method は今やクラス情報が保存されるような仕方で展開されるため，同時ドキュメント化は最早必ずしも問題にならない． (対応する情報は依然 checkDocStyle が返すオブジェクト中に含まれる．)

checkReplaceFuns はパッケージの R コード中の置き換え関数又は S3/S4 置き換えメソッドが名前が value の最終引数を持つかどうかをチェックする．

checkS3methods はパッケージの R コード中で定義された全ての S3 メソッドが対応する総称的関数の全ての引数を，メソッドに対して同じ位置の総称的関数の位置引数と共に持つかどうかをチェックする．一つの例外として，公式メソッドの最初の引数はたとえそれが総称て的関数で使われる名前でなくとも formula と呼ばれるかもしれない． ... が含まれる場合の規則は微妙である： S3 総称的と認識される関数は本体に UseMethod への呼び出しを持つもの，内部 S3 総称的(InternalMethods を見よ)，そして S3 グループ少々的(Math を見よ)である．異なる名前の下での可能な選択適用は考慮されない．総称的関数は最初与えられたパッケージ中で探され，それから **base** パッケージ中，そして(現在)パッケージ **graphics**, **stats**, そして以前の **base** を R 1.9.0 で分割して付け加えられた **utils** パッケージ中，そしてもしインストール済みパッケージがテストされていればパッケージの 'DESCRIPTION' 依存欄中にリストされたロードされた名前空間/パッケージ中でも探される．

nonS3methods(package) は package 中の S3 メソッドに'見える'がそうではない関数名を持つ文字列ベクトルを返す． package = NULL の使用は全ての既知の例を返す．

langElts は R の "言語要素"の名前のベクトルである．これらは"非常にプリミティブ"な関数として実装されている (引数リスト無し；.Primitive("<name>") とプリントされる).

もしインストール済みのパッケージを使うのなら，パッケージの全ての R オブジェクトにアクセスする必要のあるチェックは，もしかするとパッケージの既にロードされたバージョンを取り外した後で，パッケージをロードする (**base** パッケージでない限り).

Value

この関数は検出された問題に関する情報を含むそれぞれの関数名と同じ名前のクラスのオブジェクトを返す．そうしたオブジェクト中に含まれる情報を見栄え良く表示する print メソッドがある．

Rcmd *R CMD* インタフェイス

Description

R の内部から R CMD ツールを起動する．

Usage

```
Rcmd(args, ...)
```

Arguments

args	R CMD への引数の文字列ベクトル．
...	system2 に渡される引数．

Details

system2 を使い対応するシステム命令 (現在使われている R のバージョンを使う)を起動することで R CMD への可搬的で便利なインタフェイスを提供する．

Value

system2 中の節 "Value" を見よ．

Rd2HTML *Rd* ファイルの変換

Description

これらの関数は関数 parse_Rd の出力を受け取りそれからヘルプページを作る．これらは主に内部的な使用を念頭に置いており，インタフェイスは変更される可能性がある．

Usage

```
Rd2HTML(Rd, out = "", package = "", defines = .Platform$OS.type,
        Links = NULL, Links2 = NULL,
        stages = "render", outputEncoding = "UTF-8",
        dynamic = FALSE, no_links = FALSE, fragment = FALSE,
        stylesheet = "R.css", ...)

Rd2txt(Rd, out = "", package = "", defines = .Platform$OS.type,
       stages = "render", outputEncoding = "",
       fragment = FALSE, options, ...)

Rd2latex(Rd, out = "", defines = .Platform$OS.type,
         stages = "render", outputEncoding = "ASCII",
         fragment = FALSE, ..., writeEncoding = TRUE)

Rd2ex(Rd, out = "", defines = .Platform$OS.type,
      stages = "render", outputEncoding = "UTF-8",
      commentDontrun = TRUE, commentDonttest = FALSE, ...)
```

Arguments

Rd	入力として使うファイル名又は Rd オブジェクト.
out	出力を書き出すファイル名又はコネクションオブジェクト.
package	出力中にリストするパッケージ.
defines	#ifdef テスト中で使われる文字列.
stages	どのステージ("build", "install" 又は "render") で \Sexpr マクロが実行されるべきか? 下の注意を見よ.
outputEncoding	下の‘エンコーディング’節を見よ.
dynamic	論理値：動的なヘルプシステムによる翻訳時間解決に対するリンクを設定する.
no_links	論理値：他のヘルプトピックへのハイパーリンクを抑制する． R CMD Rdconv で使われる.
fragment	論理値：Rd ファイルの断片を受け入れるか． 下の注意を見よ.
stylesheet	文字列： HTML 出力ページのヘッダ中で用いられるスタイルシートに対する URL.
Links, Links2	NULL 又は findHTMLlinks が返すようなリンクの(トピックによる)名前付き文字列ベクトル.
options	Rd2txt_options に渡されるオプションの名前付きオプションのリスト.

`...`	Rd がファイル名の時 `parse_Rd` に渡される追加パラメータ.
`writeEncoding`	非アスキーエンコーディングで \inputencoding 行が書かれるべきか?
`commentDontrun`	\dontrun 節はコメントアウトすべきか?
`commentDonttest`	\donttest 節はコメントアウトすべきか?

Details

これらの関数はヘルプドキュメントを変換する： `Rd2HTML` は HTML を， `Rd2txt` は平文テキストを， `Rd2latex` は LaTeX を生成する． `Rd2ex` は `example` と R ユーティリティにより使われる書式で例を取り出す.

各々の関数は Rd ファイルに対するファイル名を受け入れ，翻訳やチェックの前にそれを構文解析するために `parse_Rd` を使う.

引数 `Link` と `Link2` の間の違いはリンクがそれらの中で順に探されることで，従って遅延評価はもし要求されるとリンクの第二水準の捜索だけを行うことに使うことが出来る.

`Rd2latex` に対する既定は第二オプションの \enc マークアップの使用を含むASCII出力であることを注意する．これは LaTeX 中の UTF-8 の使用はバージョン '2005/12/01' 又はそれ以降を必要とするために選ばれている．そのバージョンですら UTF-8 グリフのカバーは限られている(そして Latin-1 ほど完全ではない).

`Rd2txt` は `width` で決められる幅に適当な余白を持ってテキストのパラグラフを整形する．既定はバージョン R < 2.10.0 の翻訳に近い.

Unix風システムではもしオプション `"useFancyQuotes"` が真(既定)で現在のエンコーディングが UTF-8 ならば `Rd2txt` は向き付きの引用符を使う (`sQuote` を見よ).

Windows ではもしオプション `"useFancyQuotes"` が真(既定)で現在のロケールが単バイトエンコーディング(C を除く)を使えば `Rd2txt` は向き付きの引用符を使う (`sQuote` を見よ). (東アジアロケールでは引用符は普通倍幅であり英語のテキストと一緒では見かけが悪いので，向き付き引用符は試みられない．)

`Rd2txt` による整形の様々な様相は `options` 引数で制御され，`Rd2txt_options` 関数と共に文章化されている． `options` を使ってなされた変更は一時的であり， `Rd2txt_options` を使ったものは持続性がある.

`fragment = TRUE` の時は，Rd ファイルは \Sexpr 要素や #ifdef や #ifndef を使った条件付き定義の処理無しで翻訳される．普通断片は節の内部のテキストを表すが，もし断片の最初の要素がセクションマクロならば，全ての断片が通常のソート無しに節の連続として翻訳される.

Value

これらの関数は主に翻訳ヘルプページを描き込む副作用のために実行される．それらの値は(不可視の)出力ファイル名である． `Rd2latex` に対しては出力名は属性 "latexEncoding" を与えられる，これは LaTeX の 'inputenc' パッケージと共に使われるのに適した書式のファイルのエンコーディングを与える.

エンコーディング

Rd ファイルは普通多種多様なシステム上で翻訳されることを意図しているので，非ASCII文字のエンコーディングに注意する必要がある．一般にそうしたエンコーディングは正確なエンコーディングが期待できるように 'encoding'節を使って宣言されるべきである.

出力に対しては `outputEncoding` 引数が使われる： `outputEncoding = ""` は現在のシステムの素朴なエンコーディングを選択する.

もしテキストが outputEncoding に翻訳できなければ，バイト代入が使われる(iconv を見よ)： Rd2latex と Rd2ex は警告を与える．

Note

\Sexpr マクロは Rd ファイルへの新しい追加である．これは三つの時の内の一つで実行される R コードを含む： 構築時 (パッケージソースがターボール中に構築される時)， インストール時(パッケージがインストールされる時やバイナリパッケージに構築される時)，そして翻訳時 (マンページが可読な書式に変換される時)．

例えばこのマンページは：

1. 2016-09-21 at 13:49:39 に構築され，
2. 2016-09-21 at 13:49:40 にインストールされ，そして
3. 2016-09-21 at 13:49:40 に翻訳された．

Author(s)

Duncan Murdoch, Brian Ripley

References

https://developer.r-project.org/parseRd.pdf

See Also

parse_Rd, checkRd, findHTMLlinks, Rd2txt_options.

Examples

```
## Not run:
## このページのインストールと HTML とテキスト書式への翻訳をシミュレーションする：

Rd <- file.path("src/library/tools/man/Rd2HTML.Rd")

outfile <- tempfile(fileext = ".html")
browseURL(Rd2HTML(Rd, outfile, package = "tools",
          stages = c("install", "render")))

outfile <- tempfile(fileext = ".txt")
file.show(Rd2txt(Rd, outfile, package = "tools",
          stages = c("install", "render")))

checkRd(Rd) # Rd2HTML が使うのよりも厳格なテスト

## End(Not run)
```

`Rd2txt_options` テキストのヘルプの整形オプションの設定

Description

この関数はテキストのヘルプの表示に対する様々なオプションを設定する.

Usage

```
Rd2txt_options(...)
```

Arguments

`...`	名前付きのオプションを含むリスト，又は個別の名前付き引数として渡されるオプション．現在定義されているものは以下を見よ.

Details

この関数はテキスト書式のヘルプを表示するために使われる関数 `Rd2txt` に対する様々な整形オプションを永続的に設定する．現在定義されているオプションは：

width (既定値 80)：出力ページの幅.

minIndent (既定値 10)：リスト中で使われる最少インデント量.

extraIndent (既定値 4)：入れ子のリストの各々のレベル中で使われる余分のインデント量.

sectionIndent (既定値 5)：節に対するインデントレベル.

sectionExtra (既定値 2)：入れ子の節の各々に対する余分のインデント量.

itemBullet (既定値 `"* "`，アステリスクは UTF-8 と殆どの Windows ロケールでは Unicode のブレットに置き換えられる)：項目化されたリスト中でブレットとして使われるシンボル.

enumFormat ：数え上げリスト中で項目番号を整形する関数.

showURLs (既定値 `FALSE`): \href タグを展開する際 URL を示すかどうか.

code_quote (既定値 `TRUE`)： \code や一重引用符を持つ類似物を表現するかどうか.

underline_titles 既定値 `TRUE`)：節タイトルを下線で(バックスペースを用い)表示するかどうか.

Value

引数無しで呼び出されると，全てのオプション設定のリストを返す．さもなければ，それは名前付きのオプション設定を変更し不可視で前の値を返す.

Author(s)

Duncan Murdoch

See Also

`Rd2txt`

Examples

```
# itemBullet はロケール固有である
saveOpts <- Rd2txt_options()
saveOpts
Rd2txt_options(minIndent = 4)
Rd2txt_options()
Rd2txt_options(saveOpts)
Rd2txt_options()
```

Rdiff　　*R* の出力ファイルの差分

Description

二つの R 出力ファイルを与え，ヘッダー，フッターそしてその他の違いを除いた差分を計算する．

Usage

```
Rdiff(from, to, useDiff = FALSE, forEx = FALSE,
      nullPointers = TRUE, Log = FALSE)
```

Arguments

from, to	比較されるべきファイルパス．
useDiff	結果の比較に常に diff を使うべきか?
forEx	論理値：ヘッダーを取り除くための '-Ex.Rout' ファイルに対する追加の除去．
nullPointers	論理値：表示されるポインタのアドレスを比較前に 0x00000000 に設定するか?
Log	論理値：返り値は見つかった差異のログを含むべきか?

Details

R CMD BATCH からの R の開始バナーと任意のタイミング情報は共にパッケージのロードに関する行と共に取り除かれる． UTF-8 の装飾的引用符(sQuote を見よ)と， Windows では，Windows のいわゆる'スマート引用符' は単純な引用符にマップされる．環境のアドレス，コンパイルされたバイトコード，そして 16進アドレスで表される他の風変わりなタイプ (例えば <environment: 0x12345678>) は 0x00000000 にマップされる．ファイルはそれから行毎に比較される．もし同じ行数で useDiff が偽なら，単純な diff -b 風の食い違いがプリントされる (これは後に続く空白や引き続く空白の数の差を無視する．さもなければ diff -bw が編集されたファイルに対して呼び出される． (これは全ての空白の違いを無視しようと試みる：フラグ '-w' は POSIX で要求されないが GNU, Solaris そして FreeBSD バージョンではサポートされる．)

これは圧縮されていない PDF ファイルを比較可能で，生成と修正日付を無視する．

Value

もし Log が真なら，成分 status (下を見よ)と，差異を説明するもしかすると長さがゼロの文字列 out を持つリスト.

さも無ければ状態の指示で，差異がない時そしてその時だけ 0L.

See Also

R CMD Rdiff として実行されるシェルスクリプト.

Rdindex　　*Rd* ファイルから索引テーブルを作る

Description

与えられた出力ファイルやコネクションに与えられた R のドキュメントファイルからの名前とタイトル付きの2列の索引表をプリントする．タイトルは二つの列の間に見栄え良く整形される (典型的にそれぞれ 25 と 72).

Usage

```
Rdindex(RdFiles, outFile = "", type = NULL,
        width = 0.9 * getOption("width"), indent = NULL)
```

Arguments

RdFiles	索引を作成するのに使われる Rd ファイルを指定する文字列ベクトルで，ファイルへのパスを与えるか又はパッケージのソースを持つ単一のディレクトリへのパスを与える.
outFile	コネクションか，そこへ出力する出力ファイルの名前を与える文字列.
type	索引に含められるべき Rd ファイルのドキュメントのタイプを与える文字列か，又は NULL (既定). Rd ファイルのタイプは典型的には \docType タグで指定される；もし type が "data" ならばその 唯一のキーワードが datasets である Rd ファイルが同様に含められる.
width	出力中の行の折り返しのためのターゲット列を与える正整数.
indent	第二列のインデント量を指定する正整数. width/2 以上であってはならず，既定では width/3.

Details

もし名前が適正な別名でなければ，最初の別名(又はもし無ければ空文字列)が代わりに使われる.

RdTextFilter *Rd* ファイル中のテキストの選択

Description

この関数はスペルチェック等に使用するため Rd ファイル中の全てのテキストでない部分を空白にする，

Usage

```
RdTextFilter(ifile, encoding = "unknown", keepSpacing = TRUE,
             drop = character(), keep = character(),
             macros = file.path(R.home("share"), "Rd", "macros", "system.Rd"))
```

Arguments

ifile	ファイル名やコネクションとして指定された入力ファイル名，又は parse_Rd からの "Rd" オブジェクト.
encoding	parse_Rd に渡されるエンコーディング名.
keepSpacing	テキストを元々のファイルと同じ行と欄に留めておくかどうか.
drop	捨て去る追加 Rd の節.
keep	残しておく Rd の節.
macros	構文解析時に仮定するマクロ定義．parse_Rd を見よ.

Details

この関数は Rd ファイルを構文解析し，それからそれを要素ごとに見て回る．タグ "TEXT" の付いた項目は元々のファイルに登場したのと同じ位置に保たれ，一方ファイルの他の部分は空白で置き換えられるでの， aspell 等のスペルチェックソフトがテキストだけをチェック出来，元のファイル中の位置を報告する．(もし keepSpacing が FALSE ならば，空白による置き換えはされず，テキストは元々の位置には出力されない．)

既定では，タグ \S3method, \S4method, \command, \docType, \email, \encoding, \file, \keyword, \link, \linkS4class, \method, \pkg そして \var はスキップされる．追加のタグはそれらを drop 引数中にリストすることでスキップ出来る； keep 引数中にリストされたタグはスキップされない．また R 風のコード，コメント，そしてそのままのテキストに対応する c("RCODE", "COMMENT", "VERB") タグのどれも keep したり， "TEXT" を捨て去ることが可能である.

Value

文字列ベクトルで，もし一行につき一要素でファイルに書き込まれると元々の Rd ファイルのテキスト要素を複製しているかもしれない.

Note

このフィルターは Rd ファイル中で単に単語の初めをハイライトするためにマークアップが使われている時テキスト要素を単一の単語に併合しようと試みる.

Author(s)

Duncan Murdoch

See Also

これが受け入れられる `filter` である `aspell`.

`Rdutils` *Rd* オブジェクトのためのユーティリティ

Description

Rd オブジェクト中の情報について計算するユーティリティ

Usage

```
Rd_db(package, dir, lib.loc = NULL)
```

Arguments

`package`	インストール済みのパッケージの名前である文字列.
`dir`	パッケージのルートソースディレクトリへのパスを指定する文字列. これは R のドキュメントのソース(Rd 書式の)を含むサブディレクトリ 'man' を含むべきである． package が与えられない時だけ使われる.
`lib.loc`	R ライブラリのディレクトリ名の文字列か `NULL`．既定値の `NULL` は現在知られている全てのライブラリに対応する．指定されたライブラリツリーは package を探すのに使われる.

Details

`Rd_db` はパッケージ中の全ての Rd オブジェクトの簡単なデータベースを作る．これはパッケージ中の Rd ソースファイルに `parse_Rd` を実行し，プラットホームに対する条件を処理した結果のリストである.

See Also

`parse_Rd`

Examples

```
## (インストール済みの)基本パッケージの Rd データベースを作る.
db <- Rd_db("base")

## Rd オブジェクト毎のキーワードのメタデータ.
keywords <- lapply(db, tools:::.Rd_get_metadata, "keyword")
## キーワード項目を作表化.
kw_table <- sort(table(unlist(keywords)))
## 最も頻出のもの5つ：
rev(kw_table)[1 : 5]
## "最も内容のある"物：
kw_table[kw_table == 1]
```

```
## Rd ファイル毎のコンセプトメタデータ
concepts <- lapply(db, tools:::.Rd_get_metadata, "concept")
## どれだけ多くのファイルが既に \concept メタデータを持つか?
sum(sapply(concepts, length) > 0)
## 全部で幾つの概念項目があるか?
length(unlist(concepts))
```

---

read.00Index *00Index* スタイルのファイルを読み取る

Description

'00Index' 風のファイルから項目/記述情報を読み取る．そうしたファイルは表形式に表現された記述のリストであり，現在アドオンパッケージの 'INDEX' と 'demo/00Index' ファイルに対して使われる．

Usage

```
read.00Index(file)
```

Arguments

file	データ値をそこから読み取るファイルの名前．もし指定ファイルが "" ならば，入力はキーボードから取られる(この場合入力は空白行で終了できる)．もしくは file は connection で良く，これはもし必要なら開かれ，そしてもしそうであれば関数呼び出しの最後に閉じられる．

Value

項目と説明を持つ "Item" と "Description" という名前の二つの列を持つ文字行列．

See Also

項目とそれらの記述から 00Index スタイルのファイルを作る逆操作に対しては formatDL.

---

showNonASCII 非アスキー文字をピックアップする

Description

この関数は非アスキーバイトを含む文字列ベクトルの要素をプリントし，そうしたバイトは '<fc>' のようなエスケープ列でプリントされる．

Usage

```
showNonASCII(x)

showNonASCIIfile(file)
```

Arguments

x	文字列.
file	ファイルへのパス.

Details

これは最初パッケージのファイル中の可搬性の無いテキストを検出することを助けるために書かれた.

これは非アスキー文字を含む x の全ての要素をプリントする. 要素番号が先頭に付き非アスキーバイトは

iconv(sub = "byte") を使ってハイライトされる.

Value

非アスキー文字を含む x の要素が不可視で返される.

Examples

```
out <- c(
"fa\xE7ile test of showNonASCII():",
"\\details{",
"   This is a good line",
"   This has an \xfcmlaut in it.",
"   OK again.",
"}")
f <- tempfile()
cat(out, file = f, sep = "\n")

showNonASCIIfile(f)
unlink(f)
```

---

startDynamicHelp ダイナミックな *HTML* ヘルプシステムを開始する

Description

この関数は内部的なヘルプサーバーを開始するので, 要求されれば HTML ヘルプページが変換される.

Usage

```
startDynamicHelp(start = TRUE)
```

Arguments

start	論理値：動的なヘルプページを開始するか停止するか. もし NA ならサーバーが既に実行中でなければ開始される.

Details

この関数はバイブ的な HTTP サーバーを開始する，これはループバックインタフェイス (127.0.0.1) 上で稼働する．もし options("help.ports") がゼロでない整数値のベクトルに設定されると， startDynamicHelp はこれらのポートを順に試みる；さもなければそれは未使用のものを見つけるため 10 個のランダムなポートを試みる．これは環境変数 R_DISABLE_HTTPD を空でない値に設定するか options("help.ports") を 0 に設定することで無効に出来る．

startDynamicHelp はサーバーを使うことが必要な関数により呼び出されるので，滅多にユーザからは直接に呼び出されない．

HTML ヘルプを実際に使うには options(help_type = "html") を設定されなければならないことを注意するが，これは R のインストールに対しては既定値かもしれない．

もしサーバーをが開始できないか無効であれば． help.start は利用できず HTML ヘルプに対しテキストのヘルプを与えるように要求する(警告が出る)．

使用中のブラウザはループバックインタフェイスに接続出来る必要はない：時折これは全てのインタフェイスで HTTP に対するプロキシを使うように設定されるが，これは上手く行かない – 解決策は 127.0.0.1 に対する例外を加えることである．

Value

選択されたポート番号が不可視で返される (これはもしポートが停止していれば 0 になる)．

See Also

help.start と help(help_type = "html") はもし要求されれば HTTP サーバーを開始する．

Rd2HTML はパッケージのヘルプページを変換するのに使われる．

SweaveTeXFilter　　*Sweave* ファイルから *R* コードを取り去る

Description

この関数はスペルチェックやその他の目的のために Sweave 入力ファイルからコードの断片や Noweb マークアップをブランクにする．

Usage

```
SweaveTeXFilter(ifile, encoding = "unknown")
```

Arguments

ifile	入力ファイルかコネクション．
encoding	readLines に渡されるテキストエンコーディング．

Details

この関数は Sweave 入力ファイルから Noweb マークアップやコードの断片をブランクにし LaTeX ソースを後に残すので， LaTeX を認識するスペルチェッカーがそれをチェックしエラーをそれらの元々の位置で報告することが出来る．

Value

もしファイルに書き込まれれば元々の Sweave 入力ファイルのテキスト要素を一行に一要素ずつ複製する文字列ベクトル.

Author(s)

Duncan Murdoch

See Also

これが `filter = "Sweave"` と共に使われる `aspell`.

---

testInstalledPackage　インストールされたパッケージをテストする

Description

これらの関数はインストールされたパッケージ，または全ての基本と推奨パッケージのテストを行う.

Usage

```
testInstalledPackage(pkg, lib.loc = NULL, outDir = ".",
                     types = c("examples", "tests", "vignettes"),
                     srcdir = NULL, Ropts = "", ...)

testInstalledPackages(outDir = ".", errorsAreFatal = TRUE,
                      scope = c("both", "base", "recommended"),
                      types = c("examples", "tests", "vignettes"),
                      srcdir = NULL, Ropts = "", ...)

testInstalledBasic(scope = c("basic", "devel", "both", "internet"))
```

Arguments

pkg	インストール済みパッケージの名前.
lib.loc	パッケージをその中で探すライブラリのパス. `library` を見よ.
outDir	その中に出力ファイルを書き出すディレクトリ：これは既に存在しなければならない.
types	実行されるてすとのタイプ.
errorsAreFatal	論理値：テストは最初のエラーで停止すべきか?
srcdir	`.save` ファイルを探すオプションのディレクトリ.
Ropts	例やテストを実行する際 `R CMD BATCH` に渡される ‘`-d valgrind`’ の様な追加のオプション.
...	実行されるファイルを準備する際に使われる追加の引数，例えば `commentDontrun` と `commentDonttest`.
scope	どのセットがテストされるべきか? 短縮できる.

Details

これらのテストはパッケージの例のファイルがインストール済みであることに依存している (これは既定である). もしパッケージ固有のテストが 'tests' ディレクトリに見つかるとそれらはテストできる：これらは既定ではインストールされていないが，もし `R CMD INSTALL --install-tests` が使われればインストールされる．最後に任意のビニエット中の R コードを取り出すことができ検査できる.

パッケージのテストは 'outDir' の '*pkg*-tests' 副ディレクトリ中で実行されそこに出力を残す.

`testInstalledBasic` はもしインストール済みならば基本的なテストを実行する．これは `LC_COLLATE=C` を設定して実行されるべきである：関数はこれを設定しようと試みるが全ての OS で出来るわけではない．英語ではないロケールに対しては参考結果からの食い違い減らすために環境変数 `LANGUAGE` を 'en' に `LC_TIME` を 'C' に設定することが望ましい.

Windows を除いては，もし環境変数 `TEST_MC_CORES` を 1より大きな整数に設定すると `testInstalledPackages` はその値を並列プロセスの最大数として並列でパッケージのテストを実行する.

(もし Unix ならば) 基本と推奨パッケージに対するパッケージ固有のテストは普通インストールされていないが，それを行う `make install-tests` が提供されている (基本テストと並んで).

(もし Windows ならば) 基本と推奨パッケージに対するパッケージ固有のインストールのオプションである．されていないが，現在テストは完全なレポートのためもし簡単な食い違い以上のものが見つかると Unix 風の `diff` を必要とする.

Value

成功なら `0L`，失敗なら `1L` を不可視で返す.

`texi2dvi` *LaTeX* ファイルをコンパイルする

Description

dvi または PDF ファイルを作るため全ての相互参照が解決されるまで `latex`/`pdflatex`, `makeindex` そして `bibtex` を実行する.

Usage

```
texi2dvi(file, pdf = FALSE, clean = FALSE, quiet = TRUE,
         texi2dvi = getOption("texi2dvi"),
         texinputs = NULL, index = TRUE)

texi2pdf(file, clean = FALSE, quiet = TRUE,
         texi2dvi = getOption("texi2dvi"),
         texinputs = NULL, index = TRUE)
```

Arguments

file	文字列．LaTeX ソースファイルの名前.
pdf	論理値．もし TRUE なら既定の dvi ファイルの代わりに PDF ファイルを作る (texi2dvi のコマンドラインオプション '--pdf').
clean	論理値．もし TRUE なら変換中に作られた全ての補助ファイルは取り除かれる.
quiet	論理値．エラーが起きない限り出力はない.
texi2dvi	文字列(または NULL)． TeX ファイルを dvi や PDF にコンパイルするのに使われるスクリプトやプログラム．既定 ("" や "texi2dvi" や NULL で選択される)ではパス上の 'texi2dvi' という名前のプログラムかスクリプトを探し，さもなければスクリプトを system2 呼び出しを使いエミュレートする (これは値 "emulation" で選択できる)． '詳細'節も見よ.
texinputs	NULL または LaTeX と bibtex 入力の検索パスに付け加えるパスの文字列ベクトル.
index	論理値：索引を用意すべきか?

Details

texi2pdf は通常のケース texi2dvi(pdf = TRUE) に対するラッパである.

名前にもかかわらず，これは R では LaTeX ファイルをコンパイルするために使われる，特にビニエットから作られたものや Rd2pdf スクリプト(パッケージの参照マニュアルに対して使われる)．これは '*R_HOME*/share/texmf' ディレクトリが TEXINPUTS パスにあることを保証するので， 'Sweave' と 'Rd' といった R のスタイルファイルが見つかる．使用される TeX の検索パスは最初に既存の TEXINPUTS 設定(もし未設定なら現在のディレクトリ)，それから引数 texinputs の要素，そして '*R_HOME*/share/texmf' そして最後に既定のパスである．同様に BIBINPUTS と BSTINPUTS 設定に変更が加えられる.

(Unix の場合) texi2dvi に対する既定オプションは環境変数 R_TEXI2DVICMD から設定され，それに対する既定は環境変数 TEXI2DVI から設定され，もしそれが未設定なら R がコンフィギュレーションされた時の値から選ばれる.

シェルスクリプト texi2dvi は GNU の **texinfo** の一部分である．リリースされたバージョンには幾つかの問題が見られるので，もし自分のものが正しく動作しなければ R_TEXI2DVICMD=emulation を試みよ.

(Windows の場合) Windows ではオプション "texi2dvi" は普通設定されない．MiKTeX の命令 texify.exe は texi2dvi に対して適当な選択であり，, 引数が空か NULL で texi2dvi というプログラムやスクリプトが見つからなければ R 3.2.3 以来試みられる.

MiKTeX は Sept 2015 までは texi2dvi 実行ファイルを持っていたが，我々の知る他の Windows の TeX のインストールは持っていないので，エミュレーションが例えば Windows の TeXLive インストールで使われる.

時おり索引は適正な入力であっても索引作成を失敗させる特殊文字を含んでいる (特に 'hyperref' LaTeX パッケージを使う時)．引数 index = FALSE はこれが起きた時パッケージマニュアルを作ることを許すために提供されている.

Value

不可視の NULL．現在の作業ディレクトリ中に dvi または PDF ファイルを作る副作用のために使われる (そして恐らくもし特に clean = FALSE ならば他のファイルも).

Note

Unix 風システムには texi2dvi スクリプトの様々なバージョンがあるが，極めて多くのバグが見つかっており，この R ラッパはそのうちの幾つかの応急処置を与える．

texi2dvi のバージョン 4.8 (OS X で提供される)で存在したものの一つは空白を含むパスに対しては正しく動作せず，ファイルへの絶対パスが空白を含んでも正しく動作しないことであった．

三つの可能なアプローチは皆それぞれの癖を持っている．例えば Unix 風システムでは texi2dvi スクリプトは既存の補助ファイルを全て取り除くが，他の二つのアプローチはそうしない (そしてそうしたファイルで混乱させられるかもしれない)．

もしサポートされているならば(texi2dvi 5.0 とそれ以降； MiKTeX からの texify.exe)，オプション '--max-iterations=20' が無限の再実行を避けるために使われる．

エミュレーションモードは R 3.2.3 からのみ quiet = TRUE をサポートしている．現在 clean = TRUE はこのモード中で変換が成功した時だけ掃除を行う—これはユーザにエラー時にログファイルを吟味する機会を与える．

全てのアプローチは対応する命令への完全なパスに対する環境変数 LATEX, PDFLATEX, MAKEINDEX そして BIBTEX を尊重すべきである．

Author(s)

オリジナルは Achim Zeileis であるが R-core により大幅に書き換えられた．

---

toHTML *HTML* によるオブジェクト表示

Description

この総称的関数はオブジェクトから完全な HTML ページを作る．

Usage

```
toHTML(x, ...)
## S3 method for class 'packageIQR'
toHTML(x, ...)
## S3 method for class 'news_db'
toHTML(x, ...)
```

Arguments

x	表示するオブジェクト．
...	メソッドに対するオプションパラメータ； "packageIQR" と "news_db" メソッドはこれらを HTMLheader に渡す．

Value

オブジェクト x を表示する文字列ベクトル． "packageIQR" メソッドはリストを R のヘルプシステムで表示するようにデザインされている．

See Also

HTMLheader

Examples

```
cat(toHTML(demo(package = "base")), sep = "\n")
```

---

tools-deprecated　　パッケージ **tools** 中の廃止予定のオブジェクト

Description

ここにリストされる関数や変数は R の古いバージョンとの互換性のためだけに提供されており，次回のリリースで廃止されるかもしれない.

See Also

Deprecated, Defunct

---

toRd　　*Generic function to convert object to a fragment of Rd code.*

Description

この関数に対するメソッドはそれらに関連するクラスを Rd コードの断片に変換する，これはそれから text, HTML または LaTeX に変換することが出来る.

Usage

```
toRd(obj, ...)
## S3 method for class 'bibentry'
toRd(obj, style = NULL, ...)
```

Arguments

obj	変換されるオブジェクト.
style	bibentry オブジェクトの変換で使われるスタイル.
...	メソッドが使う追加引数.

Details

スタイルに関する議論は bibstyle を見よ. 既定の style = NULL 値は既定のスタイルを与える.

Value

構文解析し変換できる Rd コードの断片を含む文字列ベクトルを返す. 既定のメソッドは obj をモード character に変換し，それからその中の Rd マークアップをエスケープする. bibentry メソッドはそのクラスのオブジェクトを文献で使用するのに適当なマークアップに変換する.

toTitleCase *Convert Titles to Title Case*

Description

文字列ベクトルをタイトルケース化する，特にパッケージのタイトル.

Usage

```
toTitleCase(text)
```

Arguments

text	文字列ベクトル.

Details

これは英語のテキスト用だけを意図している.

'タイトルケース'の普遍的に受け入れられている定義はない：'主要'な単語は大文字で始まり 'for' の様な通常の単語はしないことは皆が合意するが，どの単語がそれぞれのカテゴリーに入るかについては合意がない.

一般に全て大文字の単語はそのままにされる：この移植は慣例的な 'LaTeX' と 'OpenBUGS' のような大小文字のミックスと普通大文字化されない 'jar' と 'xls' のような幾つかの技術用語を知っている．しかしながら未知の技術用語は一重引用符で囲まれた単一の単語で無い限り先頭が大文字化される：パッケージとライブラリの名前はタイトルでは引用化されるべきである.

Value

text と同じ長さの名前の無い文字列ベクトル.

---

undoc ドキュメント化されていないオブジェクトを見つける

Description

ユーザに見えるが(またはパッケージが提供するデータオブジェクトか S4 クラス)がドキュメント項目が存在しないという意味でドキュメント化されていないパッケージ中のオブジェクト見つける.

Usage

```
undoc(package, dir, lib.loc = NULL)
```

Arguments

package	インストール済みパッケージの名前の文字列ベクトル.
dir	パッケージのルートソースディレクトリへのパスを指定する文字列. これは R のドキュメントのソース(Rd 書式の)副ディレクトリ 'man' と R コードまたはデータをそれぞれ含む 'R' か 'data' 副ディレクトリの少なくとも一つを含む必要がある.
lib.loc	R ライブラリのディレクトリ名の文字列ベクトルか NULL. 既定値の NULL は現在知られている全てのライブラリに対応する. 指定されたライブラリツリーは package を探すのに使われる.

Details

この関数はパッケージの管理者に最も有用である. 原則として 全てのユーザレベルの R オブジェクトはドキュメント化されるべきである.

base パッケージは特別で, プリミティブなオブジェクトを含みそしてこれらはコードレベルで利用できる定義を持たない. 様々な目的のために使われる **base** パッケージ中の環境 .ArgsEnv と .GenericArgsEnv 中に同値なクロージャを提供している : undoc("base") は言語構成子ではない全てのプリミティブがこれらの環境中にあり他のものは無いことをチェクする.

Value

クラス "undoc" のオブジェクトで, ドキュメントのタイプで分割されたドキュメント化されていないオブジェクトの名前を含む文字列ベクトルのりスト.

そうしたオブジェクトに含まれる情報を見栄え良く表示する print メソッドがある.

See Also

codoc, QC

Examples

```
undoc("tools")                  # 'tools' 中のドキュメント化されていないオブジェクト
```

---

update_pkg_po　　パッケージに対する翻訳を準備する

Description

パッケージの 'po' ディレクトリを準備し, そして翻訳をコンパイルしインストールする.

Usage

```
update_pkg_po(pkgdir, pkg = NULL, version = NULL, copyright, bugs)
```

Arguments

pkgdir	パッケージのディレクトリへのパス.
pkg	パッケージ名：もし NULL ならばそれはパッケージの 'DESCRIPTION' ファイルから読み取られる.
version	パッケージのバージョン：もし NULL ならばそれはパッケージの 'DESCRIPTION' ファイルから読み取られる.
copyright, bugs	テンプレートファイル中の 'Copyright' と 'Report-Msgid-Bugs-To' の詳細に対するオプションの文字列.

Details

これはパッケージ中のメッセージの準備や更新を行う何段階かのステップを実行する.

- もしパッケージソースがまだ 'po' ディレクトリを持たなければ，それが作られる.
- xgettext2pot がパッケージ中の翻訳可能なメッセージを含むファイル 'po/R-*pkgname*.pot' を作ったり更新したりするために呼び出される.
- ディレクトリ po 中の名前が 'R-*lang*.po' の全てのファイルは 'R-*pkgname*.pot' から更新され， checkPoFile が更新されたファイルに対して呼び出され，そしてもし問題が何m無ければコンパイルされ 'inst/po' の下にインストールされる.
- UTF-8 ロケールでは. '翻訳' 'R-en@quot.po' は UTF-8 の向き付き引用符付きで作られ，'inst/po' の下にコンパイルしインストールされる.
- 残りのステップはファイル 'po/*pkgname*.pot' が既に存在する時だけ実行される. パッケージ中の 'src/*.{c,cc,cpp,m,mm}' ファイルが C/C++ ファイル中の翻訳可能なメッセージを含むファイル 'po/*pkgname*.pot' を作るために調べられる. もし src/windows ディレクトリがあれば，その中のファイルも調べられる.
- ディレクトリ po 中の名前が '*lang*.po' の全ての存在するファイルが '*pkgname*.pot' から更新され， checkPoFile が更新されたファイルに対して呼び出され，そしてもし何の問題も無ければファイルはコンパイルされ 'inst/po' の下にインストールされる.
- UTF-8 ロケールでは. '翻訳' 'R-en@quot.po' は UTF-8 の向き付き引用符付きで作られ，'inst/po' の下にコンパイルしインストールされる.

C/C++ のメッセージはソースファイル中で明確にマークされる必要があるので自動的には翻訳の準備がされないことを注意する. それがなされるとパッケージソース中の空のファル 'po/*pkgname*.pot' が作られこの関数を再び実行する.

pkg = "base" は特別である(そして R の開発者だけが使用する)： C ファイルはパッケージのディレクトリ中には無く，メインのソース中にある.

システム要件

この関数は GNU の gettext-tools からの以下のツールを必要とする：xgettext, msgmerge, msgfmt, msginit そして msgconv. これらはほとんどの Linux ディストリビューションの一部でありそして Unix 風システム(OS X を含む)ではソースから容易にコンパイルできる. Windows に対する予めコンパイルされたバージョンは https://www.stats.ox.ac.uk/pub/Rtools/goodies/gettext-tools.zip から入手できる.

これは UTF-8 ロケール中を除けば en@quot 翻訳に対しては恐らく正しく動作しないので，これらは他のケースではスキップされる.

See Also

xgettext2pot.

---

vignetteDepends　　ビニエットに対する従属性情報を検索する

Description

ビニエット名前を与えてビニエットが依存するパッケージに関する情報を報告する DependsList オブジェクトを作る.

Usage

```
vignetteDepends(vignette, recursive = TRUE, reduce = TRUE,
                local = TRUE, lib.loc = NULL)
```

Arguments

vignette	ビニエットソースへのパス.
recursive	間接的な従属性を含めるべきか.
reduce	全ての従属性のセットを最少の値へ縮小すべきか.
local	ローカルにだけ探索すべきか.
lib.loc	どのライブラリをローカルに探索すべきか.

Details

もし recursive が TRUE なら，従属すると指定された任意のパッケージは次にそれに従属するものを含む (そして云々)，これらは間接的な従属性と呼ばれる. , もし recursive が FALSE なら，ビニエットにより直接名指されている従属性だけが使われる.

もし local が TRUE なら，システムはユーザのローカルな機械だけを調べ，オンラインでは従属性を調べない.

もし reduce が TRUE なら，システムは DependsList オブジェクト中の欄を縮小し従属性の最少セットが指定されるようにする (例えばもし 'foo, foo (>= 1.0.0), foo (>= 1.3.0)' があれば，'foo (>= 1.3.0)' だけが返される.

Value

クラス "DependsList" のオブジェクト.

Author(s)

Jeff Gentry

See Also

pkgDepends

Examples

```
## これは lattice を必要とするのでインストールされないかもしれない
gridEx <- system.file("doc", "grid.Rnw", package = "grid")
vignetteDepends(gridEx)
```

---

vignetteEngine　　ビニエットを処理するエンジンを設定する，得る

Description

ビニエットは普通 Sweave で処理されるが，パッケージの作者は異なったエンジンを使うことを選択するかもしれない (例えば **knitr**, **noweb** または **R.rsp** パッケージが提供するもの)．この関数はこれらのパッケージによりそれらのエンジンを登録するために使われ，そして内部的に R によりそれらを検索するために使われる．

Usage

```
vignetteEngine(name, weave, tangle, pattern = NULL,
       package = NULL, aspell = list())
```

Arguments

name	エンジンの名前．
weave	ビニエットのソースファイルを LaTeX 出力に変換する関数．
tangle	ビニエットのソースファイルを R コードに変換する関数．
pattern	このエンジンで処理されるファイル名に対する正規表現パターン，もしくは既定のパターンに対しては NULL．
package	パッケージを登録するパッケージ．既定ではこれは vignetteEngine を呼び出すパッケージ．
aspell	ビニエットのソースファイル中のテキストを aspell でスペルチェックする時に使われる引数 filter と/または control をそれぞれ与える名前の要素のリスト，

Details

もし weave が無ければ，vignetteEngine は name と package にマッチする現在登録されているエンジンを返す．

もし weave が NULL なら指定されたエンジンは削除される．

他の設定は新しいエンジンを定義する． weave と tangle 関数は function(file, ...) と両立する引数リストで定義されなければならない．現在 ... 引数は論理値引数 quiet と文字列引数 encoding を含むことが出来る：これらは Sweave と Stangle に対するドキュメント中で解説されている．

weave と tangle 関数は作成された出力ファイルのファイル名を返さなければならない．現在 weave 関数は名前が '<name><pattern>' のファイルに適用すると '<name>[.](tex|pdf|html)' という名前のファイルを返す．ファイル '.tex' は pdflatex で処理されユーザに表示するための '.pdf' 出力を作成する：他は作成された様に表示される． tangle 関数はビニエットからの R コードを含む名前が '<name>[.][rRsS]' を作る． tangle 関数は引数

split = TRUE をサポートするかもしれず，そしてそれは名前が '<name>.*[.][rRsS]' のファイルを作るべきである．

pattern 引数はビニエット入力ファイルとして処理されるべきファイルの拡張子にマッチする正規表現を与える．もし NULL に設定されると，既定のパターン "[.][RrSs](nw|tex)$" が使われる．

Value

もしエンジンが削除されていれば NULL．さもなければ次の成分を含むリスト

name	エンジンの名前
package	そのパッケージの名前
pattern	ビニエット入力ファイルのパターン
weave	weave 関数
tangle	tangle 関数

Author(s)

Duncan Murdoch と Henrik Bengtsson.

See Also

Sweave そしてマニュアル 'Writing R Extensions'.

Examples

```
str(vignetteEngine("Sweave"))
```

---

write_PACKAGES *PACKAGES* ファイルを作る

Description

ソースや Mac/Windows バイナリパッケージの保存場所として 'PACKAGES' と 'PACKAGES.gz' ファイルを作る．

Usage

```
write_PACKAGES(dir = ".", fields = NULL,
               type = c("source", "mac.binary", "win.binary"),
               verbose = FALSE, unpacked = FALSE, subdirs = FALSE,
               latestOnly = TRUE, addFiles = FALSE)
```

Arguments

dir	'PACKAGES' と 'PACKAGES.gz' ファイルをそこから作り書き込み生成するレポジトリ (ソースまたはバイナリパッケージをを含むディレクトリ)の位置を記述する文字列ベクトル.
fields	既定以外に 'PACKAGES' と 'PACKAGES.gz' ファイル中で使われる欄を与える文字列ベクトル，または NULL (既定). 既定は available.packages: "Package", "Version", "Priority", "Depends", "Imports", "LinkingTo", "Suggests", "Enhances", "OS_type", "License" そして "Archs" が必要とする欄に対応し，そしてこれらは常に含められ，そしてもし addFile = TRUE ならば "File" 中にファイル名，もし副ディレクトリが使われるならば副ディレクトリへのパスが欄 "Path" 中に加わる.
type	パッケージのタイプ：現在ソースアーカイブ '.tar.{gz,bz2,xz}'，と OS X または Windows バイナリパッケージ (それぞれ '.tgz' または '.zip')がサポートされる. 既定では Windows では "win.binary"で，他では "source".
verbose	論理値. パッケージを処理中に表示すべきか?
unpacked	論理値で，パッケージの中身がパックされない形式で利用できるかまたはそうでないか(既定)を指示する.
subdirs	論理値(副ディレクトリを再帰的に含むべきかどうかを指示する) または含めるべき副ディレクトリの名前の文字列ベクトル.
latestOnly	論理値：もしパッケージ複数のバージョンが利用できるなら最新のバージョンだけを含めるべきか?
addFiles	論理値：ファイル名を 'PACKAGES' ファイル中の欄 'File' に含めるべきか.

Details

write_PACKAGES は R パッケージに対する名前付きディレクトリをスキャンし，各パッケージの 'DESCRIPTION' ファイルを取り出し，そしてこの情報を 'PACKAGES' と 'PACKAGES.gz' ファイルに書き込む.

最新でないバージョンのパッケージを含めることはそれらがより制約的でないバージョン要求を持つ時だけ役に立つので，例えば latestOnly = FALSE は 'foo_1.0' が 'R >= 2.15.0' に依存するが 'R >= 2.11.0' に依存する 'foo_0.9' が利用できる時のソースレポジトリに対して使うことが出来る.

副ディレクトリを持つ，そして従って subdirs != FALSE に対するレポジトリに対するサポートは 'PACKAGES' ファイル中の "Path" 欄の記録に依存する.

'PACKAGES' ファイル中の "File" 欄経由のより一般的なファイル名(例えば他のタイプの圧縮) が download.packages で使用できる. もしファイル名が標準的な形式でなければ addFiles = TRUE を使う.

type = "win.binary" は与えられたディレクトリ dir 中の Windows に対する(zip 圧縮された) バイナリパッケージ中に含まれる全ての 'DESCRIPTION' ファイルを読むために unz コネクションを使い，そしてこの情報からファイル 'PACKAGES' と 'PACKAGES.gz' を構築する.

Value

結果の 'PACKAGES' と 'PACKAGES.gz' ファイル中で記述されたパッケージの数を不可視で返す. もし 0 ならパッケージは見つからず如何なるファイルも書き込まれない.

Note

'DESCRIPTION' ファイルを取り出すための '.tar.gz' アーカイブの処理は非常に遅い.

この関数は他の OS から Windows の稼働する機械が利用するためのレポジトリを用意するために有用でありえるので, type = "win.binary" は全ての OS で動作すべきである.

Author(s)

Uwe Ligges と R-core.

See Also

'DESCRIPTION' ファイルの読み取りと 'PACKAGES' そして 'PACKAGES.gz' ファイルの書き込みに対しては read.dcf と write.dcf を見よ.

Examples

```
## Not run:
write_PACKAGES("c:/myFolder/myRepository")  # Windows 上
write_PACKAGES("/pub/RWin/bin/windows/contrib/2.9",
               type = "win.binary")  # Linux 上

## End(Not run)
```

---

xgettext　　パッケージ中の *R* ファイルから翻訳可能なメッセージを取り出す

Description

パッケージの(システム固有の副ディレクトリを含む) 'R' ディレクトリ中の各ファイルに対して, stop, warning, message, gettext そして gettextf, または ngettext に渡されるユニークな引数を取り出す.

Usage

```
xgettext(dir, verbose = FALSE, asCall = TRUE)

xngettext(dir, verbose = FALSE)

xgettext2pot(dir, potFile, name = "R", version, bugs)
```

Arguments

dir	ソースパッケージのディレクトリ.
verbose	論理値：各ファイルは処理中にリストすべきか?
asCall	論理値：もし TRUE なら各引数は全体として返される, さもなければ各引数中の文字列が取り出される.
potFile	作られる po テンプレートファイルの名前. 既定は 'R-*pkgname*.pot' で, ここで *pkgname* は 'dir' の基本名.
name, version, bugs	テンプレートファイル中に記録されるもの： version は既定で現在稼働中の R のバージョン番号で, bugs は "bugs.r-project.org".

Details

先頭と末尾に続く空白(スペース，タブ，そしてラインフィード)は gettext, gettextf, stop, warning そして message に対する呼び出しでは，翻訳に対する文字列を渡す内部コードによるものなので, 取り除かれる.

これらの関数が domain = NA と共に呼び出されるかを調べ，そしてもしそうなら asCall = TRUE なら呼び出しを省略する：呼び出しはもし asCall = FALSE なら可視である gettext への呼び出しを含むかもしれないことを注意する.

xgettext2pot は xgettext をそしてそれから xngettext を呼び出し， PO テンプレートファイルを **GNU Gettext** ツールと共に使うために書き出す．これは単純な翻訳に対する文字列がユニーク(**GNU Gettext** が要求するように) であることを保証するが， ngettext 呼び出しに対してはそれを行わない (そして規則は Gettext のマニュアルには述べられていないが， msgfmt はセットの間に重複があると文句を言う).

base パッケージに適用されると，これは更に '*R_HOME*/share/R' 中の '.R' ファイルも調べる.

Value

xgettext に対してはクラス "xgettext" のリスト (print メソッドを持つ)で，ソースファイル毎に翻訳可能な文字列を可能性として含む.

xngettext に対してはクラス "xngettext" のオブジェクトのリストで，それ自体長さ2の文字列のリストである.

See Also

xgettext2pot() を呼び出す update_pkg_po().

Examples

```
## Not run: ## R のソースディレクトリビルド中で：
xgettext(file.path(R.home(), "src", "library", "splines"))

## End(Not run)
```

Index

∗Topic **character**
delimMatch, 26
∗Topic **connection**
read.00Index, 56
∗Topic **datasets**
charsets, 10
∗Topic **documentation**
bibstyle, 6
buildVignette, 8
buildVignettes, 9
checkRd, 15
checkTnF, 18
checkVignettes, 19
codoc, 22
getVignetteInfo, 32
HTMLheader, 33
HTMLlinks, 34
loadRdMacros, 36
parse_Rd, 42
parseLatex, 41
QC, 46
Rd2HTML, 48
Rd2txt_options, 51
Rdindex, 53
RdTextFilter, 54
Rdutils, 55
startDynamicHelp, 57
SweaveTeXFilter, 58
toHTML, 62
tools-package, 2
toRd, 63
undoc, 64
vignetteEngine, 68
∗Topic **error**
assertCondition, 4
∗Topic **file**
fileutils, 29
read.00Index, 56
write_PACKAGES, 69
∗Topic **misc**
tools-deprecated, 63
∗Topic **package**
tools-package, 2
∗Topic **programming**
assertCondition, 4
checkFF, 11
tools-package, 2
∗Topic **utilities**
.print.via.format, 3
add_datalist, 4
buildVignettes, 9
check_packages_in_dir, 20
checkFF, 11
checkMD5sums, 13
checkPoFiles, 13
checkRdaFiles, 17
checkTnF, 18
checkVignettes, 19
compactPDF, 24
dependsOnPkgs, 26
encoded_text_to_latex, 27
getDepList, 31
getVignetteInfo, 32
HTMLheader, 33
installFoundDepends, 35
loadRdMacros, 36
make_translations_pkg, 38
makevars, 37
md5sum, 38
package.dependencies, 39
package_dependencies, 40
parse_Rd, 42
parseLatex, 41
pskill, 44
QC, 46
Rcmd, 47
Rdiff, 52
Rdindex, 53
RdTextFilter, 54
Rdutils, 55
showNonASCII, 56
SweaveTeXFilter, 58
testInstalledPackage, 59
texi2dvi, 60
toHTML, 62
toRd, 63

update_pkg_po, 65
vignetteDepends, 67
vignetteEngine, 68
write_PACKAGES, 69
xgettext, 71
∗Topic **utility**
psnice, 45
∗Topic **utilties**
bibstyle, 6
.C, *12*
.Fortran, *12*
.Platform, *29*
.libPaths, *27*, *34*
.print.via.format, 3

add_datalist, 4
Adobe_glyphs *(*charsets*)*, 10
array, *3*
as.character.Rd *(*parse_Rd*)*, 42
aspell, *54, 55*, *59*, *68*
assertCondition, 4
assertError *(*assertCondition*)*, 4
assertWarning *(*assertCondition*)*, 4
available.packages, *39, 40*, *70*

bibentry, *6, 7*, *43*, *63*
bibstyle, 6, *63*
buildVignette, 8
buildVignettes, *9*, 9

charset_to_Unicode *(*charsets*)*, 10
charsets, 10
check_packages_in_dir, 20
check_packages_in_dir_changes
*(*check_packages_in_dir*)*, 20
checkDocFiles *(*QC*)*, 46
checkDocStyle *(*QC*)*, 46
checkFF, 11
checkMD5sums, 13, *39*
checkPoFile, *66*
checkPoFile *(*checkPoFiles*)*, 13
checkPoFiles, 13
checkRd, 15, *50*
checkRdaFiles, 17
checkReplaceFuns *(*QC*)*, 46
checkS3methods *(*QC*)*, 46
checkTnF, 18
checkVignettes, 19
cite, *7*
codoc, 22, *65*
codocClasses *(*codoc*)*, 22
codocData *(*codoc*)*, 22
compactPDF, 24

conditionMessage, *5*
connection, *56*

data, *4*
Defunct, *63*
delimMatch, 26
deparseLatex *(*parseLatex*)*, 41
dependsOnPkgs, 26, *40*
Deprecated, *63*
download.packages, *70*

emptyenv, *36*
encoded_text_to_latex, 27
example, *49*

file.info, *30*
file.path, *30*
file_ext *(*fileutils*)*, 29
file_path_as_absolute *(*fileutils*)*, 29
file_path_sans_ext *(*fileutils*)*, 29
fileutils, 29
find_gs_cmd, 30
findHTMLlinks, *48*, *50*
findHTMLlinks *(*HTMLlinks*)*, 34
Foreign, *12*
format, *3*, *7*, *12*
format.compactPDF *(*compactPDF*)*, 24
formatDL, *56*

getBibstyle *(*bibstyle*)*, 6
getDepList, 31
gettext, *71*
gettextf, *71*
getVignetteInfo, 32
GSC *(*find_gs_cmd*)*, 30

help, *58*
help.ports *(*startDynamicHelp*)*, 57
help.start, *58*
hexmode, *11*
HTMLheader, 33, *62*
HTMLlinks, 34

iconv, *28*, *50*, *57*
install.packages, *35, 36*
installed.packages, *27*, *32*
installFoundDepends, *32*, 35
InternalMethods, *47*
invisible, *3*

langElts *(*QC*)*, 46
latexToUtf8 *(*parseLatex*)*, 41
library, *59*
list, *3*

list.files, *30*
list_files_with_exts *(fileutils)*, 29
list_files_with_type *(fileutils)*, 29
loadPkgRdMacros, *43*
loadPkgRdMacros *(loadRdMacros)*, 36
loadRdMacros, 36, *43*

make_translations_pkg, 38
makevars, 37
makevars_site *(makevars)*, 37
makevars_user *(makevars)*, 37
Math, *47*
mclapply, *21*
md5sum, *13*, 38
message, *71*

NativeSymbolInfo, *11*
ngettext, *71*
nonS3methods *(QC)*, 46
noquote, *11*
normalizePath, *29*

package.dependencies, *31*, 39, *40*
package_dependencies, *21*, 40
parLapply, *21*
parse_Rd, *15*, *17*, *36, 37*, 42, *48–50*, *54, 55*
parseLatex, 41
pkgDepends, *36*, *67*
pkgDepends *(getDepList)*, 31
pkgVignettes, *33*
pkgVignettes *(buildVignettes)*, 9
print, *3*, *21*
print.checkDocFiles *(QC)*, 46
print.checkDocStyle *(QC)*, 46
print.checkFF *(checkFF)*, 11
print.checkReplaceFuns *(QC)*, 46
print.checkS3methods *(QC)*, 46
print.checkTnF *(checkTnF)*, 18
print.checkVignettes *(checkVignettes)*, 19
print.codoc *(codoc)*, 22
print.codocClasses *(codoc)*, 22
print.codocData *(codoc)*, 22
print.default, *3*
print.Rd *(parse_Rd)*, 42
print.undoc *(undoc)*, 64
prompt, *23*
promptClass, *23*
pskill, 44, *46*
psnice, *45*, 45

QC, *24*, 46, *65*

R_DISABLE_HTTPD *(startDynamicHelp)*, 57
R_GSCMD *(find_gs_cmd)*, 30
R_HOME, *11*, *61*, *72*
Rcmd, 47
Rd2ex *(Rd2HTML)*, 48
Rd2HTML, *17*, *43*, 48, *58*
Rd2latex *(Rd2HTML)*, 48
Rd2pdf, *61*
Rd2txt, *51*
Rd2txt *(Rd2HTML)*, 48
Rd2txt_options, *48–50*, 51
Rd_db *(Rdutils)*, 55
Rdconv, *48*
Rdiff, 52
Rdindex, 53
RdTextFilter, 54
Rdutils, 55
read.00Index, 56
read.dcf, *71*
readLines, *58*
regexpr, *26*
resaveRdaFiles, *25*
resaveRdaFiles *(checkRdaFiles)*, 17

save, *17*
showNonASCII, 56
showNonASCIIfile *(showNonASCII)*, 56
SIGCHLD *(pskill)*, 44
SIGCONT *(pskill)*, 44
SIGHUP *(pskill)*, 44
SIGINT *(pskill)*, 44
SIGKILL *(pskill)*, 44
signalCondition, *5*
SIGQUIT *(pskill)*, 44
SIGSTOP *(pskill)*, 44
SIGTERM *(pskill)*, 44
SIGTSTP *(pskill)*, 44
SIGUSR1 *(pskill)*, 44
SIGUSR2 *(pskill)*, 44
source, *19*
sprintf, *14*
sQuote, *49*, *52*
Stangle, *8*, *19*, *68*
startDynamicHelp, 57
stop, *5*, *71*
summarize_check_packages_in_dir_depends *(check_packages_in_dir)*, 20
summarize_check_packages_in_dir_results *(check_packages_in_dir)*, 20
summarize_check_packages_in_dir_timings *(check_packages_in_dir)*, 20
summary, *21*
Sweave, *8, 9*, *19*, *68, 69*
SweaveTeXFilter, 58

Sys.getpid, *44*
system2, *47*

TEST_MC_CORES *(*testInstalledPackage*)*, 59
testInstalledBasic
        *(*testInstalledPackage*)*, 59
testInstalledPackage, 59
testInstalledPackages
        *(*testInstalledPackage*)*, 59
texi2dvi, *8, 9*, 60
texi2pdf, *8*, *10*, *19*
texi2pdf *(*texi2dvi*)*, 60
toHTML, 62
tools *(*tools-package*)*, 2
tools-deprecated, 63
tools-package, 2
toRd, 63
toTitleCase, 64
tryCatch, *5*

undoc, *24*, 64
unz, *70*
update.packages, *38*, *40*
update_pkg_po, *14*, 65, *72*

vignetteDepends, 67
vignetteEngine, 68

warning, *5*, *71*
write.dcf, *71*
write_PACKAGES, 69

xgettext, *14*, 71
xgettext2pot, *66, 67*
xgettext2pot *(*xgettext*)*, 71
xngettext *(*xgettext*)*, 71