List Methods for S3 Generic Functions or Classes

Description

List all available methods for a S3 and S4 generic function, or all methods for an S3 or S4 class.

Usage

```
methods(generic.function, class)
.S3methods(generic.function, class, envir=parent.frame())
```

Arguments

`generic.function`	a generic function, or a character string naming a generic function.
`class`	a symbol or character string naming a class: only used if `generic.function` is not supplied.
`envir`	the environment in which to look for the definition of the generic function, when the generic function is passed as a character string.

Details

`methods()` finds S3 and S4 methods associated with either the `generic.function` or `class` argument. Methods are found in all packages on the current `search()` path. `.S3methods()` finds only S3 methods, `.S4methods()` finds only only S4 methods.

When invoked with the `generic.function` argument, the `print` method displays the signatures (full names) of S3 and S4 methods. S3 methods are printed by pasting the generic function and class together, separated by a '.', as `generic.class`. The S3 method name is followed by an asterisk * if the method definition is not exported from the package namespace in which the method is defined. S4 method signatures are printed as `generic,class-method`; S4 allows for multiple dispatch, so there may be several classes in the signature `generic,A,B-method`.

When invoked with the `class` argument, the `print` method displays the names of the generic functions associated with the class, `generic`.

The source code for all functions is available. For S3 functions exported from the namespace, enter the method at the command line as `generic.class`. For S3 functions not exported from the namespace, see `getAnywhere` or `getS3method`. For S4 methods, see `getMethod`.

Help is available for each method, in addition to each generic. For interactive help, use the documentation shortcut ? with the name of the generic and tab completion, `?"generic<tab>"` to select the method for which help is desired.

The S3 functions listed are those which *are named like methods* and may not actually be methods (known exceptions are discarded in the code).

Value

An object of class `"MethodsFunction"`, a character vector of method names with `"byclass"` and `"info"` attributes. The `"byclass"` attribute is a logical(1) vector with value TRUE when the results were obtained with argument `class` defined. The `"info"` attribute is a data frame with columns:

generic

: character(), the name of the generic.

visible

: logical(), is the column exported from the namespace of the package in which it is defined?

isS4

: logical(), true when the method is an S4 method.

from

: factor(), the location or package name where the method was found.

Note

The original `methods` function was written by Martin Maechler.

References

Chambers, J. M. (1992) *Classes and methods: object-oriented programming in S.* Appendix A of *Statistical Models in S* eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.

See Also

`S3Methods`, `class`, `getS3method`.

For S4, `getMethod`, `showMethods`, `Methods`.

Examples

```
require(stats)

methods(summary)
methods(class = "aov")    # S3 class
methods("[[")             # uses C-internal dispatching
methods("$")
methods("$<-")            # replacement function
methods("+")              # binary operator
methods("Math")           # group generic
require(graphics)
methods("axis")           # looks like a generic, but is not

if(require(Matrix)) {
print(methods(class = "Matrix"))  # S4 class
m <- methods("dim")       # S3 and S4 methods
print(m)
print(attr(m, "info"))    # more extensive information

## --> help(showMethods) for related examples
}
```
