

Package ‘checkglobals’

May 8, 2026

Type Package

Title Static Analysis of R-Code Dependencies

Version 0.1.4

Date 2025-09-30

Description

A minimal R-package to approximately detect global and imported functions or variables from R-source code or R-packages by static code analysis.

BugReports <https://github.com/JorisChau/checkglobals/issues>

URL <https://jorischau.github.io/checkglobals/>,
<https://github.com/JorisChau/checkglobals>

Depends R (>= 4.1.0)

Suggests cli, knitr, jsonlite

Encoding UTF-8

License MIT + file LICENSE

Language en-US

RoxygenNote 7.3.2

NeedsCompilation yes

Author Joris Chau [aut, cre]

Maintainer Joris Chau <joris.chau@openanalytics.eu>

Repository CRAN

Date/Publication 2025-10-01 10:50:02 UTC

Contents

as.character.checkglobals	2
as.character.checkglobalsg	3
as.character.checkglobalsi	4
as.data.frame.checkglobals	5
as.data.frame.checkglobalsg	6

as.data.frame.checkglobals	7
as.matrix.checkglobals	8
as.matrix.checkglobalsg	9
as.matrix.checkglobalsi	10
as_sarif_json	11
as_sarif_json.checkglobals	12
as_vector	13
as_vector.checkglobals	14
as_vector.checkglobalsg	15
as_vector.checkglobalsi	16
checkglobals	17
check_pkg	18
check_source	20
print.checkglobals	21
print.checkglobalsg	23
print.checkglobalsi	24

Index 27

as.character.checkglobals

Cast to character

Description

Cast an S3-object of class "checkglobals" to a character vector.

Usage

```
## S3 method for class 'checkglobals'
as.character(x, pattern, which = c("global", "import"), ...)
```

Arguments

- | | |
|---------|---|
| x | object inheriting from class "checkglobals". |
| pattern | an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions. |
| which | a character vector, either "global" to print all unrecognized global variables, "import" to print all detected imported functions and variables, or both (default). |
| ... | additional arguments to configure the output: <ul style="list-style-type: none"> all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE. sorted, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE. |

Value

a character vector containing the names of the global or imported function/variables.

See Also

[as.character](#)

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
as.character(chk)

## include only imports
as.character(chk, which = "import")
```

as.character.checkglobalsg

Cast to character

Description

Cast an S3-object of class "checkglobalsg" to a character vector.

Usage

```
## S3 method for class 'checkglobalsg'
as.character(x, pattern, ...)
```

Arguments

x	object inheriting from class "checkglobalsg".
pattern	an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
...	additional arguments to configure the output: <ul style="list-style-type: none"> • all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE. • sorted, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.

Value

a character vector similar to [as.character.checkglobals](#).

See Also

[as.character.checkglobals](#)

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
as.character(chk$globals)
```

as.character.checkglobals

Cast to character

Description

Cast an S3-object of class "checkglobals" to a character vector.

Usage

```
## S3 method for class 'checkglobals'
as.character(x, pattern, ...)
```

Arguments

x	object inheriting from class "checkglobals".
pattern	an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
...	additional arguments to configure the output: <ul style="list-style-type: none">• all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE.• sorted, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.

Value

a character vector similar to [as.character.checkglobals](#).

See Also

[as.character.checkglobals](#)

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
as.character(chk$imports)
```

```
as.data.frame.checkglobals
```

Cast to data.frame

Description

Cast an S3-object of class "checkglobals" to a data.frame.

Usage

```
## S3 method for class 'checkglobals'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  pattern,
  which = c("global", "import"),
  ...
)
```

Arguments

x	object inheriting from class "checkglobals".
row.names	currently not used, included for compatibility with as.data.frame generic.
optional	currently not used, included for compatibility with as.data.frame generic.
pattern	an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
which	a character vector, either "global" to print all unrecognized global variables, "import" to print all detected imported functions and variables, or both (default).
...	additional arguments to configure the output: <ul style="list-style-type: none"> • all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE. • sorted, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.

Value

a data.frame with three character columns:

- name, the name of the global or imported function/variable.
- package, the import package, only applies to imported functions/variables.
- type, the type of the detected entity, either "global" or "import".

See Also

[as.data.frame](#)

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
as.data.frame(chk)

## include only selected imports
as.data.frame(chk, pattern = "coef", which = "import")
```

```
as.data.frame.checkglobalsg
      Cast to data.frame
```

Description

Cast an S3-object of class "checkglobalsg" to a data.frame.

Usage

```
## S3 method for class 'checkglobalsg'
as.data.frame(x, row.names = NULL, optional = FALSE, pattern, ...)
```

Arguments

x	object inheriting from class "checkglobalsg".
row.names	currently not used, included for compatibility with as.data.frame generic.
optional	currently not used, included for compatibility with as.data.frame generic.
pattern	an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
...	additional arguments to configure the output:

- `all.names`, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a `'.'` are omitted. Defaults to TRUE.
- `sorted`, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.

Value

a data.frame similar in format to [as.data.frame.checkglobals](#).

See Also

[as.data.frame.checkglobals](#)

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
as.data.frame(chk$globals)
```

as.data.frame.checkglobals

Cast to data.frame

Description

Cast an S3-object of class `"checkglobals"` to a data.frame.

Usage

```
## S3 method for class 'checkglobals'
as.data.frame(x, row.names = NULL, optional = FALSE, pattern, ...)
```

Arguments

<code>x</code>	object inheriting from class <code>"checkglobals"</code> .
<code>row.names</code>	currently not used, included for compatibility with as.data.frame generic.
<code>optional</code>	currently not used, included for compatibility with as.data.frame generic.
<code>pattern</code>	an optional regular expression . Only names matching <code>pattern</code> are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
<code>...</code>	additional arguments to configure the output: <ul style="list-style-type: none"> • <code>all.names</code>, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a <code>'.'</code> are omitted. Defaults to TRUE. • <code>sorted</code>, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.

Value

a data.frame similar in format to `as.data.frame.checkglobals`.

See Also

`as.data.frame.checkglobals`

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
as.data.frame(chk$imports)
```

`as.matrix.checkglobals`

Cast to matrix

Description

Cast an S3-object of class "checkglobals" to a matrix.

Usage

```
## S3 method for class 'checkglobals'
as.matrix(x, pattern, which = c("global", "import"), ...)
```

Arguments

<code>x</code>	object inheriting from class "checkglobals".
<code>pattern</code>	an optional regular expression . Only names matching <code>pattern</code> are returned. <code>glob2rx</code> can be used to convert wildcard patterns to regular expressions.
<code>which</code>	a character vector, either "global" to print all unrecognized global variables, "import" to print all detected imported functions and variables, or both (default).
<code>...</code>	additional arguments to configure the output: <ul style="list-style-type: none"> <code>all.names</code>, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a <code>'.'</code> are omitted. Defaults to TRUE. <code>sorted</code>, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.

Value

a character matrix with three columns:

- name, the name of the global or imported function/variable.
- package, the import package, only applies to imported functions/variables.
- type, the type of the detected entity, either "global" or "import".

See Also

[as.matrix](#)

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
as.matrix(chk)

## include only selected imports
as.matrix(chk, pattern = "coef", which = "import")
```

as.matrix.checkglobalsg

Cast to matrix

Description

Cast an S3-object of class "checkglobalsg" to a matrix.

Usage

```
## S3 method for class 'checkglobalsg'
as.matrix(x, pattern, ...)
```

Arguments

x	object inheriting from class "checkglobalsg".
pattern	an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
...	additional arguments to configure the output: <ul style="list-style-type: none"> • all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE. • sorted, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.

Value

a matrix similar in format to [as.matrix.checkglobals](#).

See Also

[as.matrix.checkglobals](#)

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
as.matrix(chk$globals)
```

as.matrix.checkglobals

Cast to matrix

Description

Cast an S3-object of class "checkglobals" to a matrix.

Usage

```
## S3 method for class 'checkglobals'
as.matrix(x, pattern, ...)
```

Arguments

x	object inheriting from class "checkglobals".
pattern	an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
...	additional arguments to configure the output: <ul style="list-style-type: none">• all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE.• sorted, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.

Value

a matrix similar in format to [as.matrix.checkglobals](#).

See Also

[as.matrix.checkglobals](#)

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
as.matrix(chk$imports)
```

as_sarif_json

Cast to SARIF json

Description

as_sarif_json is a generic function to cast objects returned by [checkglobals](#), [check_pkg](#) or [check_source](#) to SARIF (Static Analysis Results Interchange Format) json for upload to external CI tools (e.g. GitHub Code Scanning, Jenkins warnings-ng plugin or Azure DevOps). The function invokes particular *methods* which depend on the [class](#) of the first argument.

Usage

```
as_sarif_json(x, path, pattern, which, ...)
```

Arguments

x	an S3-object to convert.
path	file path to write the SARIF json content to.
pattern	an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
which	a character vector, either "global" to include all unrecognized global variables, "import" to include all detected imported functions and variables, or both (default).
...	additional arguments to configure the returned output.

Value

json object.

```
as_sarif_json.checkglobals
    Cast to SARIF json
```

Description

Cast an S3-object of class "checkglobals" to SARIF json.

Usage

```
## S3 method for class 'checkglobals'
as_sarif_json(x, path, pattern, which = c("global", "import"), ...)
```

Arguments

x	object inheriting from class "checkglobals".
path	(optional) file path to write the SARIF json content. If missing, the SARIF content is returned as an object of class json.
pattern	an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
which	a character vector, either "global" to print all unrecognized global variables, "import" to print all detected imported functions and variables, or both (default).
...	additional arguments to configure the output: <ul style="list-style-type: none"> • all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE. • pretty, a logical value passed to <code>jsonlite::toJSON</code>. Defaults to FALSE. • root_dir, directory to use as root relative to which all result uri's are generated. • use_cli, a logical value indicating if cli should be used to format the result messages. Defaults to TRUE, which means that cli-formatting is attempted if cli is installed. • markdown, a logical value indicating if markdown result fields should be included. Defaults to FALSE. • first_only, a logical value to include only the first detected location of an unknown global variable or function (instead of all detected locations). Defaults to FALSE.

Value

if path is provided writes the SARIF json content to path and returns the path invisibly, otherwise returns the SARIF json content as a json object..

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
json <- as_sarif_json(chk, pretty = TRUE, markdown = TRUE)
```

as_vector

Cast to list vector generic

Description

as_vector is a generic function to cast objects returned by [checkglobals](#), [check_pkg](#) or [check_source](#) to list vectors. The function invokes particular *methods* which depend on the [class](#) of the first argument.

Usage

```
as_vector(x, pattern, which, ...)
```

Arguments

x	an S3-object to convert.
pattern	an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
which	a character vector, either "global" to print all unrecognized global variables, "import" to print all detected imported functions and variables, or both (default).
...	additional arguments to configure the returned output.

Value

a list of character vectors.

as_vector.checkglobals

Cast to list vector

Description

Cast an S3-object of class "checkglobals" to a list vector.

Usage

```
## S3 method for class 'checkglobals'  
as_vector(x, pattern, which = c("global", "import"), ...)
```

Arguments

x	object inheriting from class "checkglobals".
pattern	an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
which	a character vector, either "global" to print all unrecognized global variables, "import" to print all detected imported functions and variables, or both (default).
...	additional arguments to configure the output: <ul style="list-style-type: none">• all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE.• sorted, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.

Value

a list consisting of three character vectors:

- global, vector of global function/variable names.
- import, vector of import function/variable names.
- package, vector of import package names.

Examples

```
## R-package from folder  
chk <- checkglobals(  
  pkg = system.file(  
    "unit_tests", "pkg", "testpkg",  
    package = "checkglobals"  
  )  
)  
as_vector(chk)  
  
## include only selected imports  
as_vector(chk, pattern = "coef", which = "import")
```

```
as_vector.checkglobalsg
```

Cast to list vector

Description

Cast an S3-object of class "checkglobalsg" to a list vector.

Usage

```
## S3 method for class 'checkglobalsg'  
as_vector(x, pattern, ...)
```

Arguments

x	object inheriting from class "checkglobalsg".
pattern	an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
...	additional arguments to configure the output: <ul style="list-style-type: none">• all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE.• sorted, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.

Value

a list consisting of one character vector:

- global, vector of global function/variable names.

See Also

[as_vector.checkglobals](#)

Examples

```
## R-package from folder  
chk <- checkglobals(  
  pkg = system.file(  
    "unit_tests", "pkg", "testpkg",  
    package = "checkglobals"  
  )  
)  
as_vector(chk$globals)
```

`as_vector.checkglobals`*Cast to list vector*

Description

Cast an S3-object of class "checkglobals" to a list vector.

Usage

```
## S3 method for class 'checkglobals'  
as_vector(x, pattern, ...)
```

Arguments

<code>x</code>	object inheriting from class "checkglobals".
<code>pattern</code>	an optional regular expression . Only names matching <code>pattern</code> are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
<code>...</code>	additional arguments to configure the output: <ul style="list-style-type: none">• <code>all.names</code>, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE.• <code>sorted</code>, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.

Value

a list consisting of two character vectors:

- `import`, vector of import function/variable names.
- `package`, vector of import package names.

See Also

[as_vector.checkglobals](#)

Examples

```
## R-package from folder  
chk <- checkglobals(  
  pkg = system.file(  
    "unit_tests", "pkg", "testpkg",  
    package = "checkglobals"  
  )  
)  
as_vector(chk$imports)
```

 checkglobals

Check R-source code or R-packages for globals and imports

Description

Approximately detect global and imported functions or variables from R-scripts, folders, R-code strings or R-packages by static code analysis. This function is simply a convenience wrapper around [check_source](#) and [check_pkg](#) and the return value is the same as calling these functions directly. If called without a file, dir, text or pkg argument, the function is run in the current working directory. If the current working directory is an R-package folder, this is identical to `checkglobals(pkg = ".")`, otherwise the behavior is the same as `checkglobals(dir = ".")`.

Usage

```
checkglobals(..., include_compiled = FALSE, skip_globals = NULL)
```

Arguments

`...` can be any one of the following arguments:

- `file`, file character path to R-script to analyze, can be either a file on the local filesystem or a remote file location (e.g. a server or the web).
- `text`, character R-code string to analyze.
- `dir`, character path to folder with R-scripts to analyze.
- `pkg`, character path to R-package, can be either:
 - a local R-package folder;
 - path to bundled (tar.gz) R-package on local filesystem;
 - remote path to bundled (tar.gz) R-package, (e.g. a remote server or the web).

`include_compiled` logical value indicating if compiled functions called with `.Call`, `.C`, `.External`, `.Fortran` should be included as global variables.

`skip_globals` optional character vector of names to skip/exclude as (unrecognized) global variables.

Value

list S3-object of class "checkglobals" with three components:

- `globals`, list of class "checkglobalsg"
- `imports`, list of class "checkglobalsi"
- `missing_packages`, character vector with missing packages

for programmatic use, cast the returned S3-object with `as.data.frame`, `as.matrix`, `as.character` or `as_vector`.

See Also

[check_source](#), [check_pkg](#)

Examples

```
## local R-script
checkglobals(
  file = system.file(
    "unit_tests", "pkg", "testpkg", "R", "functions1.R",
    package = "checkglobals"
  )
)

## local R-folder
checkglobals(
  dir = system.file(
    "unit_tests", "pkg", "testpkg", "R",
    package = "checkglobals")
)

## R-code string
checkglobals(text = "cc <- function(m) stats::coef(m)")

## R-package from folder
checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)

## R-script from remote location

checkglobals(
  file = "https://raw.githubusercontent.com/rstudio/shiny-examples/main/004-mpg/app.R"
)

## R-package from remote location

check_pkg(
  pkg = "https://cran.r-project.org/src/contrib/tinytest_1.4.1.tar.gz",
  skip_globals = "cluster"
)
```

Description

Approximately detect global and imported functions or variables from R-packages by static code analysis. Conceptually, the function inspects all files in the package R-folder and contrasts the detected (unrecognized) globals and imports against the imports listed in the NAMESPACE of the R-package. R-scripts present elsewhere in the package (i.e. not in the R-folder) are not scanned, as these are not coupled to the package NAMESPACE file.

Usage

```
check_pkg(pkg = ".", include_compiled = FALSE, skip_globals = NULL)
```

Arguments

pkg	character path to R-package, can be either: <ul style="list-style-type: none">• a local R-package folder;• path to bundled (tar.gz) R-package on local filesystem;• remote path to bundled (tar.gz) R-package, (e.g. a remote server or the web).
include_compiled	logical value indicating if compiled functions called with <code>.Call</code> , <code>.C</code> , <code>.External</code> , <code>.Fortran</code> should be included as global variables.
skip_globals	optional character vector of names to skip/exclude as (unrecognized) global variables.

Value

list S3-object of class "checkglobals" with three components:

- globals, list of class "checkglobalsg"
- imports, list of class "checkglobalsi"
- missing_packages, character vector with missing packages

for programmatic use, cast the returned S3-object with `as.data.frame`, `as.matrix`, `as.character` or `as_vector`.

See Also

[checkglobals](#), [check_source](#)

Examples

```
## from R-package folder
check_pkg(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
```

```
## from bundled R-package

check_pkg(
  pkg = "https://cran.r-project.org/src/contrib/tinytest_1.4.1.tar.gz"
)
```

check_source

Check R-scripts, folders or R-code strings for globals and imports

Description

Approximately detect global and imported functions or variables from R-scripts, folders or R-code strings by static code analysis. For inspection of individual R-scripts use the ‘file’ argument, for R-code strings use the ‘text’ argument, and for folders containing R-scripts use the ‘dir’ argument. This function does not require executing the code under inspection.

Usage

```
check_source(file, text, dir, include_compiled = FALSE, skip_globals = NULL)
```

Arguments

file	character path to R-script to analyze, can be either a file on the local filesystem or a remote file location (e.g. a server or the web).
text	character R-code string to analyze.
dir	character path to folder with R-scripts to analyze.
include_compiled	logical value indicating if compiled functions called with <code>.Call</code> , <code>.C</code> , <code>.External</code> , <code>.Fortran</code> should be included as global variables.
skip_globals	optional character vector of names to skip/exclude as (unrecognized) global variables.

Value

list S3-object of class "checkglobals" with three components:

- `globals`, list of class "checkglobalsg"
- `imports`, list of class "checkglobalsi"
- `missing_packages`, character vector with missing packages

for programmatic use, cast the returned S3-object with `as.data.frame`, `as.matrix`, `as.character` or `as_vector`.

See Also

[checkglobals](#), [check_pkg](#)

Examples

```
## local R-script
check_source(
  file = system.file(
    "unit_tests", "pkg", "testpkg", "R", "functions1.R",
    package = "checkglobals"
  )
)

## local R-folder
check_source(
  dir = system.file(
    "unit_tests", "pkg", "testpkg", "R",
    package = "checkglobals"
  )
)

## R-code string
check_source(text = "cc <- function(m) stats::coef(m)")

## R-script from remote location

check_source(
  file = "https://raw.githubusercontent.com/rstudio/shiny-examples/main/004-mpg/app.R"
)
```

```
print.checkglobals      Print "checkglobals" object
```

Description

Print method for S3-objects of class "checkglobals" as returned by [checkglobals](#), [check_pkg](#) or [check_source](#). Prints the *name* and *location* of all unrecognized global variables; and the *name* and *location* of all detected imported functions grouped by R-package. The *location* consists of the source file name and line number. If [cli](#) is installed and cli-hyperlinks are supported in the console, clicking the *location* links opens the source file at the given line number. The bars printed behind the import package names are filled based on the absolute number of detected imports per package.

Usage

```
## S3 method for class 'checkglobals'
print(
  x,
  format = c("basic", "detail"),
  pattern,
  which = c("global", "import"),
  ...
)
```

Arguments

x	object inheriting from class "checkglobals".
format	character, one of the following two choices: <ul style="list-style-type: none"> • "basic", (default) prints only the name and source code location of the detected globals and imports. • "detail", prints the name and location of the detected globals and imports, as well as the lines in the source code file comprising the detected globals and imports. The maximum number of lines printed per source code reference can be specified using <code>maxLines</code>.
pattern	an optional regular expression . Only names matching <code>pattern</code> are returned. <code>glob2rx</code> can be used to convert wildcard patterns to regular expressions.
which	a character vector, either "global" to print all unrecognized global variables, "import" to print all detected imported functions and variables, or both (default).
...	additional arguments to configure the printed output. The following arguments can be specified: <ul style="list-style-type: none"> • <code>all.names</code>, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE. • <code>includePkgs</code>, a character vector of imported package names. Only the imports for these packages are printed. Defaults to NA, which includes all detected packages in the printed output. • <code>maxRef</code>, the maximum number of printed source code references per detected global/import. Defaults to 1. • <code>maxLines</code>, the maximum number of printed lines per source code reference, only used if <code>format = "detail"</code>. Defaults to 5. • <code>use_cli</code>, a logical value indicating if <code>cli</code> should be used to format the printed output. Defaults to TRUE, which means that <code>cli</code>-formatting is attempted if <code>cli</code> is installed. • <code>maxWidth</code>, the maximum column width of the printed output. If <code>cli</code> is installed, the default width is determined by <code>cli::console_width()</code>. If <code>cli</code> is not installed, <code>getOption("width")</code> is checked. If <code>getOption("width")</code> is undefined, the column width defaults to 80.

Value

Returns the object `x` *invisibly* (via [invisible](#)).

See Also

[checkglobals](#), [check_pkg](#), [check_source](#)

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
```

```

    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
chk

## print globals with references to source code
print(chk, format = "detail", which = "global", maxRef = 99)

## print selected imports
print(chk, format = "detail", pattern = "coef", which = "import", maxRef = 99)

## print without cli-formatting
print(chk, use_cli = FALSE)

```

```
print.checkglobalsg Print "checkglobalsg" object
```

Description

Print method for S3-objects of class "checkglobalsg" characteristic to the "globals" list element of "checkglobals" objects returned by [checkglobals](#), [check_pkg](#) or [check_source](#).

Usage

```
## S3 method for class 'checkglobalsg'
print(x, format = "basic", pattern, ...)
```

Arguments

x	object inheriting from class "checkglobalsg".
format	character, one of the following two choices: <ul style="list-style-type: none"> "basic", (default) prints only the name and source code location of the detected globals. "detail", prints the name and location of the detected globals, as well as the lines in the source code file comprising the detected globals. The maximum number of lines printed per source code reference can be specified using maxLines.
pattern	an optional regular expression . Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
...	additional arguments to configure the printed output. The following arguments can be specified: <ul style="list-style-type: none"> all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE.

- `includePkgs`, a character vector of imported package names. Only the imports for these packages are printed. Defaults to `NA`, which includes all detected packages in the printed output.
- `maxRef`, the maximum number of printed source code references per detected global/import. Defaults to 1.
- `maxLines`, the maximum number of printed lines per source code reference, only used if `format = "detail"`. Defaults to 5.
- `use_cli`, a logical value indicating if `cli` should be used to format the printed output. Defaults to `TRUE`, which means that `cli`-formatting is attempted if `cli` is installed.
- `maxWidth`, the maximum column width of the printed output. If `cli` is installed, the default width is determined by `cli::console_width()`. If `cli` is not installed, `getOption("width")` is checked. If `getOption("width")` is undefined, the column width defaults to 80.

Value

Returns the object `x` invisibly (via `invisible`)

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
chk$globals

## print globals with references to source code
print(chk$globals, format = "detail", maxRef = 99)

## print without cli-formatting
print(chk$globals, use_cli = FALSE)
```

print.checkglobals *Print "checkglobals" object*

Description

Print method for S3-objects of class `"checkglobals"` characteristic to the `"imports"` list element of `"checkglobals"` objects returned by `checkglobals`, `check_pkg` or `check_source`.

Usage

```
## S3 method for class 'checkglobals'
print(x, format = "basic", pattern, ...)
```

Arguments

x	object inheriting from class "checkglobals".
format	character, one of the following two choices: <ul style="list-style-type: none"> • "basic", (default) prints only the name and source code location of the detected imports. • "detail", prints the name and location of the detected imports, as well as the lines in the source code file comprising the detected imports. The maximum number of lines printed per source code reference can be specified using <code>maxLines</code>.
pattern	an optional regular expression . Only names matching <code>pattern</code> are returned. <code>glob2rx</code> can be used to convert wildcard patterns to regular expressions.
...	additional arguments to configure the printed output. The following arguments can be specified: <ul style="list-style-type: none"> • <code>all.names</code>, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE. • <code>includePkgs</code>, a character vector of imported package names. Only the imports for these packages are printed. Defaults to NA, which includes all detected packages in the printed output. • <code>maxRef</code>, the maximum number of printed source code references per detected global/import. Defaults to 1. • <code>maxLines</code>, the maximum number of printed lines per source code reference, only used if <code>format = "detail"</code>. Defaults to 5. • <code>use_cli</code>, a logical value indicating if <code>cli</code> should be used to format the printed output. Defaults to TRUE, which means that <code>cli</code>-formatting is attempted if <code>cli</code> is installed. • <code>maxWidth</code>, the maximum column width of the printed output. If <code>cli</code> is installed, the default width is determined by <code>cli::console_width()</code>. If <code>cli</code> is not installed, <code>getOption("width")</code> is checked. If <code>getOption("width")</code> is undefined, the column width defaults to 80.

Value

Returns the object `x` *invisibly* (via [invisible](#))

Examples

```
## R-package from folder
chk <- checkglobals(
  pkg = system.file(
    "unit_tests", "pkg", "testpkg",
    package = "checkglobals"
  )
)
chk$imports

## print selected imports
print(chk, format = "detail", pattern = "coef", which = "import", maxRef = 99)
```

```
## print without cli-formatting  
print(chk$imports, use_cli = FALSE)
```

Index

.C, 17, 19, 20
.Call, 17, 19, 20
.External, 17, 19, 20
.Fortran, 17, 19, 20

as.character, 3, 17, 19, 20
as.character.checkglobals, 2, 3, 4
as.character.checkglobalsg, 3
as.character.checkglobalsi, 4
as.data.frame, 5–7, 17, 19, 20
as.data.frame.checkglobals, 5, 7, 8
as.data.frame.checkglobalsg, 6
as.data.frame.checkglobalsi, 7
as.matrix, 9, 17, 19, 20
as.matrix.checkglobals, 8, 10, 11
as.matrix.checkglobalsg, 9
as.matrix.checkglobalsi, 10
as_sarif_json, 11
as_sarif_json.checkglobals, 12
as_vector, 13, 17, 19, 20
as_vector.checkglobals, 14, 15, 16
as_vector.checkglobalsg, 15
as_vector.checkglobalsi, 16

check_pkg, 11, 13, 17, 18, 18, 20–24
check_source, 11, 13, 17–19, 20, 21–24
checkglobals, 11, 13, 17, 19–24
class, 11, 13

glob2rx, 2–16, 22, 23, 25

invisible, 22, 24, 25

print.checkglobals, 21
print.checkglobalsg, 23
print.checkglobalsi, 24

regular expression, 2–16, 22, 23, 25