

# Package: checkglobals (via r-universe)

August 26, 2024

**Type** Package

**Title** Static Analysis of R-Code Dependencies

**Version** 0.1.2

**Date** 2024-03-29

**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

**Encoding** UTF-8

**License** MIT + file LICENSE

**Language** en-US

**RoxygenNote** 7.2.0

**Repository** <https://jorischau.r-universe.dev>

**RemoteUrl** <https://github.com/jorischau/checkglobals>

**RemoteRef** HEAD

**RemoteSha** f0d830907e41529bff8aceffc12cea22020202a3

## 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.checkglobalsi . . . . .	7

as.matrix.checkglobals	8
as.matrix.checkglobalsg	9
as.matrix.checkglobalsi	10
as_vector	11
as_vector.checkglobals	12
as_vector.checkglobalsg	13
as_vector.checkglobalsi	14
checkglobals	15
check_pkg	16
check_source	18
print.checkglobals	19
print.checkglobalsg	21
print.checkglobalsi	22

## Index 24

---

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 <a href="#">regular expression</a> . Only names matching pattern 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 output: <ul style="list-style-type: none"> <li>• <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.</li> <li>• <code>sorted</code>, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.</li> </ul>

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

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

## 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 <a href="#">as.data.frame</a> generic.
optional	currently not used, included for compatibility with <a href="#">as.data.frame</a> generic.
pattern	an optional <a href="#">regular expression</a> . Only names matching pattern are returned. <a href="#">glob2rx</a> 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"> <li>• all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE.</li> <li>• sorted, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.</li> </ul>

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

**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 <a href="#">regular expression</a> . 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"> <li><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.</li> <li><code>sorted</code>, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.</li> </ul>



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

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

**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\_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 <a href="#">regular expression</a> . Only names matching pattern are returned. <a href="#">glob2rx</a> 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 <a href="#">regular expression</a> . Only names matching pattern are returned. <a href="#">glob2rx</a> 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"><li>• all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE.</li><li>• sorted, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.</li></ul>

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

## 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 <a href="#">regular expression</a> . Only names matching <code>pattern</code> are returned. <a href="#">glob2rx</a> can be used to convert wildcard patterns to regular expressions.
<code>...</code>	additional arguments to configure the output: <ul style="list-style-type: none"><li>• <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.</li><li>• <code>sorted</code>, a logical value indicating if the function/variable names should be sorted alphabetically. Defaults to TRUE.</li></ul>

### 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"> <li>• a local R-package folder;</li> <li>• path to bundled (tar.gz) R-package on local filesystem;</li> <li>• remote path to bundled (tar.gz) R-package, (e.g. a remote server or the web).</li> </ul>
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",
  skip_globals = "cluster"
)
```

---

 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"> <li>• "basic", (default) prints only the name and source code location of the detected globals and imports.</li> <li>• "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>.</li> </ul>
pattern	an optional <a href="#">regular expression</a> . Only names matching pattern 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"> <li>• <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.</li> <li>• <code>maxRef</code>, the maximum number of printed source code references per detected global/import. Defaults to 1.</li> <li>• <code>maxLines</code>, the maximum number of printed lines per source code reference, only used if <code>format = "detail"</code>. Defaults to 5.</li> <li>• <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.</li> <li>• <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.</li> </ul>

**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"> <li>"basic", (default) prints only the name and source code location of the detected globals.</li> <li>"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.</li> </ul>
pattern	an optional <a href="#">regular expression</a> . Only names matching pattern are returned. <a href="#">glob2rx</a> 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"> <li>all.names, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE.</li> <li>maxRef, the maximum number of printed source code references per detected global/import. Defaults to 1.</li> <li>maxLines, the maximum number of printed lines per source code reference, only used if format = "detail". Defaults to 5.</li> </ul>

- `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:

- `"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 `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:
- `all.names`, a logical value. If TRUE, all object names are returned. If FALSE, names which begin with a '.' are omitted. Defaults to TRUE.
  - `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$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*, 15, 17, 18  
*.Call*, 15, 17, 18  
*.External*, 15, 17, 18  
*.Fortran*, 15, 17, 18

*as.character*, 3, 15, 17, 18  
*as.character.checkglobals*, 2, 3, 4  
*as.character.checkglobalsg*, 3  
*as.character.checkglobalsi*, 4  
*as.data.frame*, 5–7, 15, 17, 18  
*as.data.frame.checkglobals*, 5, 7, 8  
*as.data.frame.checkglobalsg*, 6  
*as.data.frame.checkglobalsi*, 7  
*as.matrix*, 9, 15, 17, 18  
*as.matrix.checkglobals*, 8, 10, 11  
*as.matrix.checkglobalsg*, 9  
*as.matrix.checkglobalsi*, 10  
*as\_vector*, 11, 15, 17, 18  
*as\_vector.checkglobals*, 12, 13, 14  
*as\_vector.checkglobalsg*, 13  
*as\_vector.checkglobalsi*, 14

*check\_pkg*, 11, 15, 16, 16, 18–22  
*check\_source*, 11, 15–17, 18, 19–22  
*checkglobals*, 11, 15, 17–22  
*class*, 11

*glob2rx*, 2–14, 20, 21, 23

*invisible*, 20, 22, 23

*print.checkglobals*, 19  
*print.checkglobalsg*, 21  
*print.checkglobalsi*, 22

*regular expression*, 2–14, 20, 21, 23