Package 'RcppXPtrUtils'

January 20, 2025

Type Package

Title XPtr Add-Ons for 'Rcpp'

Version 0.1.2

Description Provides the means to compile user-supplied C++ functions with 'Rcpp' and retrieve an 'XPtr' that can be passed to other C++ components.

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/Enchufa2/RcppXPtrUtils

BugReports https://github.com/Enchufa2/RcppXPtrUtils/issues

Depends R (>= 3.0.0)

Imports Rcpp

Suggests inline, testthat

RoxygenNote 7.2.0

NeedsCompilation no

Author Iñaki Ucar [aut, cph, cre] (<https://orcid.org/0000-0001-6403-5550>)

Maintainer Iñaki Ucar <iucar@fedoraproject.org>

Repository CRAN

Date/Publication 2022-05-24 19:10:32 UTC

Contents

---

checkXPtr   *Check an* XPtr*'s Signature*

Description

Check the signature (i.e., arguments and return type) of the output of [cppXPtr](), which is an external pointer wrapped in an object of class XPtr. If the user-supplied C++ function does not match the signature, the wrapper throws an informative error.

Usage

```
checkXPtr(ptr, type, args = character(), call. = TRUE)
```

Arguments

ptr   an object of class XPtr compiled with [cppXPtr]().

type   the return type.

args   a list of argument types.

call.   logical, indicating if the call should become part of the error message.

See Also

[cppXPtr]()

Examples

```
# takes time to compile
ptr <- cppXPtr("double foo(int a, double b) { return a + b; }")
checkXPtr(ptr, "double", c("int", "double")) # returns silently
try(checkXPtr(ptr, "int", c("double", "std::string"))) # throws error
```

---

cppXPtr   *Define an* XPtr *with a C++ Implementation*

Description

Dynamically define an XPtr with C++ source code. Compiles and links a shared library with bindings to the C++ function using [cppFunction](), then returns an XPtr that points to the function and can be used to be plugged into another C++ backend.

Usage

```
cppXPtr(
  code,
  depends = character(),
  plugins = character(),
  includes = character(),
  rebuild = FALSE,
  cacheDir = getOption("rcpp.cache.dir", tempdir()),
  showOutput = verbose,
  verbose = getOption("verbose")
)
```

Arguments

code	Source code for the function definition.
depends	Character vector of packages that the compilation depends on. Each package listed will first be queried for an inline plugin to determine header files to include. If no plugin is defined for the package then a header file based the package's name (e.g. PkgName.h) will be included.
plugins	Character vector of inline plugins to use for the compilation.
includes	Character vector of user includes (inserted after the includes provided by depends).
rebuild	Force a rebuild of the shared library.
cacheDir	Directory to use for caching shared libraries. If the underlying code passed to sourceCpp has not changed since the last invocation then a cached version of the shared library is used. The default value of tempdir() results in the cache being valid only for the current R session. Pass an alternate directory to preserve the cache across R sessions.
showOutput	TRUE to print R CMD SHLIB output to the console.
verbose	TRUE to print detailed information about generated code to the console.

Value

An object of class XPtr that points to the compiled function.

See Also

[cppFunction](#), [checkXPtr](#)

Examples

```
# takes time to compile
ptr <- cppXPtr("double foo(int a, double b) { return a + b; }")
class(ptr)
print(ptr)
```

Index