Package: nobcom (via r-universe)

May 24, 2026

Title Turn off the byte-compiler

Description Import this package in order to turn off R's byte-compiler

Version 1.0.4

Author Mark Bravington

Maintainer Mark Bravington<markb2@summerinsouth.net>

ByteCompile no

License GPL-2

Repository https://markbravington.r-universe.dev

Date/Publication 2025-08-27 11:55:20 UTC

RemoteUrl https://github.com/markbravington/nobcom

RemoteRef HEAD

RemoteSha 7dafe3a3cdd83c1a19a278394d10eefd8678e63e

Contents

`nobcom-package` *How to use the nobcom package*

Description

R's byte-compiler can occasionally cause problems for other packages— `offarray` v1.0.x is an example. So you might need to turn off the byte compiler *before* loading such a package. You can do that by adding

"Imports: nobcom" to the DESCRIPTION. That avoids having to get the user to remember to do something special before `library` (or, worse, before an "invisible" automatic package load occurs). Turning off the byte-compiler is harmless, with (in my experience) minimal speed penalty. Of

course, in the long run I suppose one "should" try to write code that does not offend the byte-compiler— but (i) I never asked for the byte-compiler, and (ii) "in the long run we are all dead" (JM Keynes).

The user will never directly load the **nobcom** package, nor run any function in it. All the work is done by `nobcom:::.onLoad`, which runs automatically when `nobcom` is loaded. The details are a bit tricky; best to look at the code. The point of making `nobcom` into a package, is that "Imports" to another package are loaded (I hope) *before* that package is, so there is no opportunity (I hope) for the byte-compiler to do its dirty work; if you wait until the other package's `.onLoad`, or try to set a "load hook" (`?setHook`) it's too late.

To prevent this behaviour— ie to leave the byte-compiler unaffected— set the environment variable `R_IGNORE_NOBCOM` to any non-empty string before loading "your" package. That envar is checked during the exection of `nobcom:::.onLoad`.

That's all, folks!

Index