R package dependencies not installed from Additional_repositories
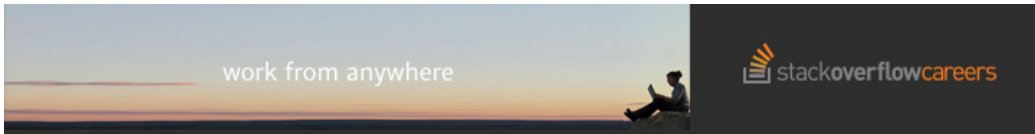
I am developing a package that requires the namespace of another package, OpenMx. This package is only available from it's own repository, but specifying this repository in the Additional_repositories field does not work. When trying to build my package without OpenMx installed, I get:

ERROR: dependency 'OpenMx' is not available for blah blah

Is there a problem with my description, or something else I need to be doing?

The relevant piece of my description file:

```
Depends: R (>= 3.0.0)
Imports:  MASS, OpenMx (>= 2.0)
Additional_repositories: http://openmx.psyc.virginia.edu/OpenMx2/
LazyData: Yes
VignetteBuilder: knitr
Suggests: knitr
```

Log from a build attempt:

```
Thu Apr  2 18:08:10 2015: Building tarball for package ctsem (SVN revision 5)
using R version 3.1.3 Patched (2015-03-16 r67994) ...

* checking for file 'ctsem/DESCRIPTION' ... OK
* preparing 'ctsem':
* checking DESCRIPTION meta-information ... OK
* installing the package to build vignettes
      ---------------------------------
* installing *source* package 'ctsem' ...
** R
** data
*** moving datasets to lazyload DB
** inst
** preparing package for lazy loading
Error in loadNamespace(i, c(lib.loc, .libPaths()), versionCheck = vI[[i]]) :
  there is no package called 'OpenMx'
ERROR: lazy loading failed for package 'ctsem'
* removing '/tmp/RtmpAHPlFq/Rinst5272759a2048/ctsem'
      ---------------------------------
ERROR: package installation failed
Run time: 1.09 seconds.
```

r   package   packages

edited Apr 8 at 11:44

asked Apr 2 at 18:18
Charlie
30 ● 6

The R package system is pretty good but with four or five possible repositories, you don't always succeed with installing packages that have multiple dependencies. I usually just look at the list of missing packages and track them down. In your case I think that may not even be an R package and you would need to use your system resources outside of R. ... Nope I was wrong. Do some Google searching. It's not hard to find the answer. – BondedDust Apr 2 at 20:53 ✎

That would be fine if I just wanted to install the package, but I'd like to make it available via R-forge and later CRAN, so it needs to pass build checks. – Charlie Apr 2 at 21:27

Then look at how other package authors handle it. cran.r-project.org/web/packages/simsem/index.html – BondedDust Apr 2 at 21:38

simsem doesn't use any OpenMx functions, as far as I can see - doesn't import the namespace. – Charlie Apr 2 at 21:45 ✎

It doesn't import the NAMESPACE because that would make it ineligible for hosting on CRAN. It probably does use a `source` call in some of its examples that are bracketed by `## Not run:` ... `## End(Not run)`. – BondedDust Apr 2 at 21:52

2 Answers

AFAIK `Additional_repositories` is still a free-form field and not automatically added to the `options("repos")` consulted by `download.packages()` and hence `install.packages()` or `update.packages()` .

For the very problem of adding *repos* beyond CRAN, I wrote a little helper package drat which a few people, myself included, use to host repos either on GitHub (a "default" mode as it is so easy via `gh-pages`) or any other http-accessible server---ie local servers within a company or workgroup. Drat does nothing particularly difficult, but for this default case of adding a repository of user, say, `openmx` , all you need is `drat::addRepo("openmx")` and the rest is inferred from defaults (and assumes a `drat` repo in account `openmx` exists).

I deploy both use cases: newer packages via repos at GitHub, and a work-local repo.

Edit: But WRE says

> The 'Addiitonal_repositories' field is a comma-separated list of repository URLs where the packages named in the other fields may be found. It is currently used by 'R CMD check' to check that the packages can be found, at least as source packages (which can be installed on any platform).

which implies it looks there. Odd. And the repo looks legit at the given URL.

Edit 2: And I sent a patch to R Core for the typo in the above quote.

edited Apr 8 at 12:20   answered Apr 8 at 11:52

                                                         Dirk Eddelbuettel
                                                         158k ● 16 ● 275 ● 382

Did you find this question interesting? Try our newsletter

Sign up for our newsletter and get our top new questions delivered to your inbox (see an example).

So it appears that the Additional_repositories field is only used to check the existence of any suggested packages, but they are not actually loaded, and hence if any examples or vignettes use these packages, the build process will break. I have worked around the problem for now by setting examples to not run and including a pre-built vignette instead.

answered Apr 23 at 15:16

Charlie
30 ● 6