

# **Computational Aspects and Windows Related Community Services on CRAN**

Uwe Ligges

Department of Statistics, TU Dortmund

useR!2009, Rennes

# Contents and Introduction

- a CRAN repository contains more than 1859 packages – as of July 04, 2009
- the R package system made R so successful:
  - ease of installing packages
  - checks help to provide at least some kind of quality
- more and more has been or has to be automated
- inverse recursive checks and rebuilds after updates
- check result summaries as useR/developER service
- win-builder service for building Windows binary packages

# Where come the Windows binaries from?

- 1 Package developer uploads a package to CRAN/incoming
- 2 Kurt checks it, and moves it to the source repository
- 3 win-builder looks up recent changes among source packages, then builds, checks and uploads to a local site – including reverse dependencies
- 4 CRAN master mirrors the win-builder's site

# Human resources

It is all automated, so what?

## Tasks of a repository maintainer

- Maintaining and adapting the scripts themselves
- Setting up repositories for new versions of R
- Handling errors that were not covered by the scripts
- Answering questions of developereRs and useRs
- Asking package maintainers to fix their packages !!!

# Computer resources

## Why should we 'improve' computer resources?

### Tasks of a CRAN auto-build-and-check machine

- Build and check new and updated packages (including reverse dependencies) on a daily basis – at least for R-release and R-devel
  - Notifications for developers / check summaries
  - Make R-devel on a daily basis
  - Re-check all packages for R-devel (and R-patched?) on a regular (weekly?) basis
- Aim:** make it possible to look out for errors for R developers and package developers
- Provide the ftp check system for package developers

# Reverse dependencies

Consider

- package *B* depends on package *A*
- packages *C* and *D* depend on package *B*,

then once *A* is updated, we definitely need to

- re-check packages *A*, *B*, *C*, *D*,

because newly introduced features or changes in *A* could have broken something somewhere else.

# Reverse dependencies

- We even may need – in addition to a new binary for *A* – some updated *binary* packages for *B*, *C* and *D*:  
e.g. if S4 classes are involved and/or saved images are involved
- Currently, this means for an update of the recommended package *MASS* (inclusive the recursive dependencies):
  - `tools::dependsOnPkgs("pkgname")`

# Reverse dependencies

- Number of packages to be rebuilt and rechecked on CRAN + BioConductor + Omegahat within Depends, Imports, Suggests, Enhances:

hwriter	<b>6</b>
relations	<b>6</b>
sets	<b>7</b>
Matrix	<b>992</b>
lme4	<b>992</b>
scatterplot3d	<b>1002</b>
MASS	<b>1026</b>

- Obviously a problem given the runtime and many package updates a day — at least without allowing for parallel checks

# Computer resources

Why should we 'improve' computer resources?

Tasks of a CRAN auto-build-and-check machine

Since it is desirable to get check results for development processes early, we do need a build/check system that

**finished within (at least!) 24 hours**

for each flavor of R in order to

- provide the check results when needed, not thereafter
- make binaries available in time during an R release cycle (e.g. the day after alpha/beta/rc/release)

# CRAN Windows Binaries' Package Check

Last updated on 2006-06-13 17:51:39 (**useR! 2006**) (simplified)

No	Package	Version	R-2.3.1	Inst. time	Check time
...	...	...	...	...	...
742	wavelets	0.2-1	OK	26	88
743	waveslim	1.5	OK	58	109
744	wavethresh	2.2-8	OK	30	75
745	wccsom	1.1.0	OK	18	87
746	wle	0.9-2	OK	24	365
747	xgobi	1.2-13	<b>ReadMe</b>		
748	xtable	1.3-2	OK	22	52
749	zicounts	1.1.4	<b>WARNING</b>	26	46
<b>750</b>	<b>zoo</b>	1.1-0	OK	23	60
SUM (in hours) on <b>Xeon 3.06 GHz:</b>				6.34	19.77
				<b>≈ 26 hours</b>	

# CRAN Windows Binaries' Package Check

Last updated on 2007-01-27 19:45:44 (**DSC 2007**) (simplified)

No	Package	Version	R-2.4.1	Inst. time	Check time
...	...	...	...	...	...
935	wle	0.9-3	OK	18	196
936	wnominate	0.93	OK	23	200
937	xgobi	1.2-13	<b>ReadMe</b>		
938	xlsReadWrite	1.1.1	OK	10	30
939	xtable	1.4-3	OK	14	55
940	yalmpute	0.0-1	OK	21	88
941	zicounts	1.1.4	<b>WARNING</b>	18	30
942	zipfR	0.6-0	OK	14	65
<b>943</b>	<b>zoo</b>	1.2-2	OK	15	53
<b>SUM (in hours) on Athlon64 X2 5000+:</b>				4.42	17.52
				<b>≈ 22 hours</b>	

# CRAN Windows Binaries' Package Check

Last updated on 2009-07-04 18:20:05 (**useR! 2009**) (simplified)

No	Package	Version	R-2.9.1	Inst. time	Check time
...	...	...	...	...	...
1851	xts	0.6-6	OK	15	38
1852	yalmpute	1.0-9	OK	20	66
1853	yacca	1.1	OK	7	18
1854	yaml	1.0.1	OK	12	17
1855	yest	0.4-1	<b>WARNING</b>	7	35
1856	zipfR	0.6-5	OK	17	50
1857	zoeppritz	1.0-2	OK	6	17
1858	zoo	1.5-6	OK	10	66
<b>1859</b>	zyp	0.9-1	OK	6	17
<b>SUM (in hours), 2x Xeon E5430 Quad:</b>				6.44/8	30.33/8
				<b>≈ 5 hours</b>	

# CRAN Windows Binaries' Package Check

Due to the requirements for the CRAN maintenance, parallel package installation has been made possible since R-2.9.0 (and has been used on CRAN before):

- package dependencies are calculated by R
- written to a Makefile, and
- finally resolved by 'make -jX'

A similar process is used for checking packages on CRAN.

# CRAN Windows Binaries' Package Check

Makefile (as used on CRAN):

```
PKG := packageA packageB packageC
PKG_INST := $(PKG:=-install.out)
all: $(PKG_INST)
%-install.out: %
    MAKE=make MAKEFLAGS= R -f install.R\  
        --vanilla --quiet --args Path/to/library\  
        build $< R_default_packages=NULL
packageA-install.out: packageB-install.out
```

# Other community services

Beside the community services such as the maintenance of CRAN repositories (thanks to Kurt Hornik), there are others:

- R-Forge, <http://r-forge.r-project.org/>, for hosting projects, incl.:
  - svn repositories
  - R package build and check facilities (on a regular basis)
  - several other features
- Winbuilder, <http://win-builder.r-project.org/>, for on demand build and check of Windows binary packages
  - intended for developers without access to Windows boxes

# Automated Windows builds and checks

shortened version of the web page, part 1

## Building and checking R source packages for Windows

- This site provides services for building and checking R source packages for 32-bit Windows.
- This service is intended for useRs who do not have Windows available for checking and building Windows binary packages. Windows useRs can easily build and check their packages on Windows with the setup described in the [R1aA] manual.
- A Windows [...] machine [...] builds and checks your source packages and makes the installation log, check log and the Windows binary available.
- Currently, both [...] R-2.9.1 and R-devel [...] are running and ready for uploads.

# Automated Windows builds and checks

shortened version of the web page, part 2

## Building and checking R source packages for Windows

The procedure is as follows:

- Prepare your source package (using R CMD build) and add yourself including your e-mail address in the 'Maintainer' field of the DESCRIPTION file [...].
- Check the package on your platform (using R CMD check).
- If the package passes the checks under your platform, upload it using passive ftp in binary mode [...] to the directory R-release [...] or R-devel [...] on `ftp://win-builder.r-project.org/`. Passive ftp can be performed with [...].

# Automated Windows builds and checks

shortened version of the web page, part 3

Building and checking R source packages for Windows

**Finally** wait for half an hour or so and check the e-mail address you have specified in the 'Maintainer' field of the package's DESCRIPTION file. An e-mail should have arrived that will explain where the binary and the log files can be downloaded from: A randomly generated directory on this server which will be deleted after roughly 72 hours.

# Automated Windows builds and checks

shortened version of the web page, part 4

## Disclaimer

- You are using this service on your own risk!
- We do not guarantee privacy nor confidentiality of the uploaded files or data in any way: Names of uploaded packages are always listable by everybody. After the compilation process we try to hide the binary package as well as the logs using randomly chosen directory names. Nevertheless, everybody who knows or guesses those names can download those files.
- Although virus checking is performed on the compiled binary packages, we do not guarantee that files made available for download are free of malware (such as viruses or trojans).
- The results of R CMD check are returned from a particular given system with a particular setup and might not be reproducible on other systems. Hence we do not guarantee for the results of R CMD check.

# Summary and Future

- parallel installation and checks of R packages (on repository and useR level)
- better checks for package dependencies (helps to improve quality of packages that (reverse) depend on others)
- community services (convenient package development and distribution)

# Summary and Future

- **unsolved**: we are currently rebuilding too much in respect to reverse dependencies (see DSC 2009)
- **future**: package installation and management can be improved by
  - some better database like system for each library (no need to parse thousands of DESCRIPTION files) / each repository
  - moving more perl code to R (as we have seen a speed gain when porting the INSTALL script from perl to R): 'Rd2 parser'