Writing CRAN Task Views

Achim Zeileis

CRAN task views should provide some guidance which part of the growing number of packages on CRAN are useful for a certain topic. They should give a brief overview of the contained packages and allow automatic installation. Infrastructure for both tasks is provided in `.ctv` files, an XML-based specification of CRAN task views. The format is based on just a handful of new tags and is mostly self-explanatory. A brief artificial example is given here, more detailed explanations are given below:

```
<CRANTaskView>

  <name>Econometrics</name>
  <topic>Computational Econometrics</topic>
  <maintainer email="Achim.Zeileis@R-project.org">Achim Zeileis</maintainer>
  <version>2008-02-15</version>

  <info>
    Emphasize something <b>important</b> with standard HTML tags,
    and reference the package <pkg>foo</pkg> and also the view
    <view>MachineLearning</view>.
  </info>

  <packagelist>
    <pkg>foo</pkg>
    <pkg priority="core">bar</pkg>
  </packagelist>

  <links>
    <a href="http://path/to/homepage/">The Title of a Relevant Homepage</a>
    <view>MachineLearning</view>
  </links>

</CRANTaskView>
```

The whole document is framed by an opening and closing `<CRANTaskView>` tag. Within these, there are six sections: `<name>`, `<topic>`, `<maintainer>`, `<info>`, `<packagelist>` and `<links>`.

`<name>` gives the name of the task view. This is used as the identifier for installing the view and as the name for the XML file, e.g., `Econometrics.ctv`, and the auto-generated HTML file, e.g., `Econometrics.html`. Hence, it should be not too long and contain no special characters like spaces.

`<topic>` is a plain text specification of the topic of the task view. (Note that in XML files there are three special characters `<`, `>` and `&` that need to be escaped by their HTML counterparts.)

`<maintainer>` gives the name of the maintainer in plain text with support for specification of an e-mail address.

`<version>` is specified by a date (in ISO format).

`<info>` should be a short description of the packages that explains which packages are useful for which tasks. It can also contain any further informations and should be written in plain HTML code, i.e., any HTML markup can be used (preferably following XHTML conventions). There are five additional tags available: `<pkg>`, `<view>`, `<bioc>`, `<ohat>`, `<rforge>`, `<gcode>`, and `<code>`. `<pkg>` and `<view>` should be used as markup for packages and views (in the same repository), respectively. During generation of the corresponding HTML file, they will be replaced by hyperlinks to the packages/views. Similarly, `<bioc>`, `<ohat>`, `<rforge>`, and `<gcode>` can be used to refer to projects on Bioconductor, Omegahat, R-Forge, and Google Code. These will be replaced by the appropriate hyperlinks to the project on `http://www.Bioconductor.org/`, `http://www.Omegahat.org/`, `http://R-Forge.R-project.org/`, and `http://Code.Google.com/`, respectively. Finally, `<code>` can be used as markup for R code which will simply be set in a typewriter font.

`<packagelist>` contains the list of packages associated with the view. It should contain (at least) all packages mentioned in the `<info>` section and allows for a distinction between `"core"` and `"normal"` (default) priority packages. The distinction is only important for the installation of certain views because the user can specify whether all packages (default) or only the most important core packages should be installed (with all their dependencies).

`<links>` is a means of specifying further places where information about the discussed topic and related issues is discussed, e.g., manuals/books/tutorials on how to do Econometrics, say, with R.

Remark: If you need to include special characters, e.g., an umlaut, in a `.ctv` file, you can do so in the `<info>` section by simply writing it in HTML code (note, however, that `&` is the escape character in XML/HTML). Otherwise, you can include special characters in UTF-8 encoding provided you add the declaration `<?xml version="1.0" encoding="UTF-8"?>` in the first line of the `.ctv` file. This applies in particular if you need an umlaut in the `<maintainer>` field.

For a template of a `.ctv` file, look at the included `Econometrics.ctv`. It can be read and printed by

```
> library("ctv")
> x <- read.ctv(file.path(.find.package("ctv"), "ctv", "Econometrics.ctv"))
> x
```

```
CRAN Task View
--------------
Name:       Econometrics
Topic:      Computational Econometrics
Maintainer: Achim Zeileis
Contact:    Achim.Zeileis@R-project.org
Version:    2009-02-25

Packages:   AER, bayesm, betareg, bivpois, boot, bootstrap, car*, CDNmoney,
            crq, Design, dlm, dse, dyn, dynlm, Ecdat, FinTS, forecasting,
            frontier, gmm, Hmisc, ineq, its, lme4, lmtest*, Matrix, meboot,
            mFilter, micEcon, mlogit, MNP, MSBVAR, nlme, np, plm*, pscl, pwt,
            quantreg, reldist, sampleSelection, sandwich*, sde, segmented, sem,
            simpleboot, SparseM, strucchange, systemfit, timeSeries, tsDyn,
            tseries*, tsfa, urca*, uroot, vars, VR, zoo*
            (* = core package)
```

The resulting object can be transformed into HTML by `ctv2html(x)`. Once this view has been installed at a CRAN-style repository as `http://CRAN.R-project.org/web/views/`, it can be installed by the user by `install.views("Econometrics")`.