`xtrcode` v.0.2 — extract contents of LaTeX environments from a file

Thomas Ruedas[*]

January 12, 2000

1 Description

This is a small Perl program I originally made when I was writing a program documentation for a large program consisting of several files; I wanted to extract the code out of the LaTeX verbatim environments containing the whole code portionwise and have it put into a file which could then be compiled directly. This is how it works:

`xtrcode` <*options*> *texfile* <*codefile*>

Options:

`-a` extract all code; this is for code which is supposed to be distributed over different target files, so do not use it together with *codefile*

`-e`***environment*** extract content of *environment*; default is [Vv]erbatim

`-p`***pattern*** marker pattern for environments to extract; default is `%%@`. A null pattern is possible.

-h show help

You can download it together with a short documentation as a tar.gz file from `http://www.geophysik.uni-frankfurt.de/~ruedas/download/xtrcode.tar.gz`. In the following I assume that *texfile* is a program documentation.

***codefile*:** `xtrcode` *texfile codefile* searches for a marker pattern followed by a blank and *codefile* and copies the content of the following environment into the target file *codefile*. This can be used to extract only the contents corresponding to a certain part of the program instead of the whole lot.—If no *codefile* is given, the default output file is "xtrcode.out".

Option -a: `xtrcode -a` *texfile* extracts all contents of an environment from *texfile* and writes it into the output file which is expected to be specified by the use of the marker pattern. This is useful if you have program code which is intended to come into separate files; when documenting it you will use a kind of extra pattern line, e.g. `%%@ code1.f` preceding each verbatim environment containing parts of `code1.f`, and similarly with the other files. With the `-a` option, `xtrcode` will interpret `code1.f` as the name of the file which shall contain the content of the following environment, and will copy the contents of all environments of the selected type into the respective target files.—The use of a marker pattern is necessary here, and no *codefile* should be specified.

Option -e: `xtrcode -e`*environment texfile* searches for *environment* instead of the default verbatim or Verbatim.

Option -p: `xtrcode -p`*pattern texfile* uses *pattern* instead of the default `%%@` as a marker. If `-p` is used without a pattern, no marker patterns are taken into account.

If there are nested environments of the same type, the outermost is assumed to be the one to extract.

[*] e-mail: `ruedas@geophysik.uni-frankfurt.de`

2 Example

Consider a LaTeX file `progdoc.tex` with the following description of two programs:

```
In the following I describe my sophisticated program which consists of a
main program and a subroutine.\par
In the main part, the subroutine is called:
%%@ main.f
\begin{verbatim}
      program main
      open(10,file="dat1",status="new")
      write (*,*) 'Enter number to be squared:'
      read (*,*) zin
      call sub1(zin,zout)
\end{verbatim}
Then the result is written to a file.
%%@ main.f
\begin{verbatim}
      write (10,*) zout
      close(10)
      end
\end{verbatim}
The subroutine calculates the result by
\begin{equation}
y=x^2.
\end{equation}
\verb/x*x/ was used instead of \verb/x**2/ to increase efficiency.
%%@ sub1.f
\begin{verbatim}
      subroutine sub1(x,y)
      y=x*x
      return
      end
\end{verbatim}
A further subroutine for calculation of cubes is under development.
```

With `xtrcode -a progdoc.tex`, all the code in the verbatim environments will be extracted, copying everything marked with `%%@ main.f` into a file `main.f` and everything marked with `%%@ sub1.f` into a file `sub1.f`. With `xtrcode progdoc.tex sub1.f` you can extract everything marked with `%%@ sub1.f`; with `xtrcode -eequation -p progdoc.tex` you can extract all lines of all equation environments.