The “multibbl” package

Apostolos Syropoulos
366, 28th October Str.
GR-671 00 Xanthi, HELLAS
Email:`apostolo@obelix.ee.duth.gr`
or
`apostolo@ocean1.ee.duth.gr`

2004/07/27

Abstract

The package “multibbl” redefines the standard commands bibliographic commands so that one can generate multiply reference sections. Each section has it own auxiliary file (for use with BibTeX) and title.

1 Introduction

This package has been designed for people who write a document that has references in languages that use different scripts. For example, if one prepares a book in Greek, then his/her document may have references in Greek and in English. In this particular case, it makes no sense to sort the references and have only one bibliography section. It is better to have two bibliography sections: one for the Greek references and on for the English references. Moreover, it is possible to have references in more than two scripts. Then we obviously, need three or even more bibliography sections. By the redefining the standard LaTeX user interface commands, this package makes it possible to have two or more bibliography sections in one document. Naturally, one can use this package for documents that use only one script, just in case there is a need to have multiply bibliography sections.

2 The code

As usual, we first have to identify the package:

```
⟨package⟩
\NeedsTeXFormat{LaTeX2e}[1996/01/01]
\ProvidesPackage{multibbl}[2007/07/27\space v1.1]
\typeout{Package: ‘multibbl’ v1.1\space <2004/07/27> (A. Syropoulos)}
```

`\newbibliography` Now we have to define a new command which has one argument: the name of a new auxiliary file. If that file does not exist, it simply creates this file.

```
\def\newbibliography#1{\begingroup
  \if@filesw \expandafter\newwrite\csname #1@auxfile\endcsname
  \expandafter\immediate\openout \csname #1@auxfile\endcsname #1.aux\relax
  \typeout{Writing auxiliary file #1.aux }\fi \endgroup}
```

`\@citex` The command `\@citex` is used internally to process the arguments of the `\cite` command. Since, the later command has now two required arguments, the name of an auxiliary file and the usual citation labels, we have to redefine the `\@citex` command so that these labels are written into the auxiliary file, just in case we want to generate the bibliography with BibTeX. What we actually do is to replace the code

```
\if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
```

with the code

```
 \@ifundefined{#2@auxfile}{}{\expandafter\immediate%
       \write\csname #2@auxfile\endcsname{\string\citation{\@citeb}}}%
```

This code checks whether we have defined the control sequence `#2@auxfile`, where `#2` is the first required argument, i.e., the name of the auxiliary file, and if it is it writes the citation labels to this file.

```
\def\@citex[#1]#2#3{%
  \let\@citea\@empty
  \@cite{\@for\@citeb:=#3\do
    {\@citea\def\@citea{,\penalty\@m\ }%
     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
     \@ifundefined{#2@auxfile}{}{\expandafter\immediate%
        \write\csname #2@auxfile\endcsname{\string\citation{\@citeb}}}%
     \@ifundefined{b@\@citeb}{\mbox{\reset@font\bfseries ?}%
       \G@refundefinedtrue
       \@latex@warning
         {Citation `\@citeb' on page \thepage \space undefined}}%
       {\hbox{\csname b@\@citeb\endcsname}}}}{#1}}
```

`\bibliography` The `\bibliography` command has been redefined to take three required arguments: the first is the name of the auxiliary file, the second is the argument that the standard command has and the third is the the title of the bibliographic section, i.e., what will appear in the running heads, etc. In order to do this, we have to redefine the commands `\bibname` and `\refname`, which are used to typeset the running heads, etc.

```
\def\bibliography#1#2#3{%
   \@ifundefined{#1@auxfile}{}{\expandafter\immediate%
     \write\csname #1@auxfile\endcsname{\string\bibdata{#2}}}%
  \def\bibname{#3}%
  \def\refname{#3}%
  \@input@{#1.bbl}}
```

`\bibliographystyle` The `\bibliographystyle` command has two arguments: the name of an auxiliary file and the name of a bibliographic style. With this command one can specify the bibliographic style each bibliography section will have. Although, at first sight it makes no sense to use different bibliographic styles. However, we believe that this feature may be proved useful in cases where it is not possible to distinguish to which bibliography section each citation belongs to.

```
\def\bibliographystyle#1#2{%
  \ifx\@begindocumenthook\@undefined\else
    \expandafter\AtBeginDocument
  \fi
  \@ifundefined{#1@auxfile}{}{\expandafter\immediate%
      \write\csname #1@auxfile\endcsname{\string\bibstyle{#2}}}}%
```

`\nocite` The `\nocite` command has been redefined so that it has two required arguments: the name of an auxiliary file and the standard argument.

```
\def\nocite#1#2{\@bsphack
  \@for\@citeb:=#2\do{%
    \edef\@citeb{\expandafter\@firstofone\@citeb}%
    \@ifundefined{#1@auxfile}{}{\expandafter\immediate%
        \write\csname #1@auxfile\endcsname{\string\citation{\@citeb}}}%
    \@ifundefined{b@\@citeb}{\G@refundefinedtrue
        \@latex@warning{Citation `\@citeb' undefined}}{}}%
  \@esphack}
```

Although it is not necessary to explicitly terminate a package, we believe it is good practice to do it.

```
\endinput
⟨/package⟩
```

Acknowledgment

I thank Antonis Tsolomitis for checking this usability of this package.

Dedication

To the fond memory of my late brother Mikhail Syropoulos.