

L^AT_EX

Footnote Strikes A Wrong Chord: How Do You Conquer It?

Jackie Damrau

Have you ever tried to change the footnote symbol? Did you find it easy? Well, the way to change the type of symbol used by the `\footnote` environment is, e.g., by:

```
\renewcommand{\thefootnote}
{\fnsymbol{footnote}}
```

which provides the *, **, †, ‡, §, ¶, © construction. Should you desire alphabetic (upper or lowercase), the constructions would be:

```
\renewcommand{\thefootnote}{\Alph{footnote}}
\renewcommand{\thefootnote}{\alph{footnote}}
```

or could conceivably be anything else you would wish to use.

Query from the College Station Meeting

Lowell Smith posed a question at the 11th Annual T_EX Users Group Meeting in College Station where he wanted to change the `\footnote` environment inside a minipage and no matter what he tried received lowercase alphabet letters for the footnote symbol.

I would like to throw this question out to the L^AT_EX community to see what responses are submitted. Any solutions submitted will be reviewed in the next column.

Earlier Column Revisited

In *TUGboat* 11, no. 1, this column presented macros for changing from single space to double space. I received a response from Josephine Colmenares at Fordham University with a simpler macro that she has given permission to use. The macros are:

```
\newcommand{\single}{%
\renewcommand{\baselinestretch}{1}
\normalsize}
\newcommand{\double}{%
\renewcommand{\baselinestretch}{1.5}
\normalsize}
```

◊ Jackie Damrau
 Computer Operations Support
 MS-1011
 SSC Laboratory
 2550 Beckleymeade Avenue
 Dallas, TX 75237-3946
 (214) 708-6048
 Internet: damrau@sscvx1.ssc.gov

A L^AT_EX Document Style Option for Typesetting APL

Andreas Geyer-Schulz, Josef Matulka and
 Gustaf Neumann

Abstract

In this article we describe the L^AT_EX document style option `apl.sty` for typesetting documents containing passages in APL code. All symbols needed within a multi-vendor APL environment are provided. Currently the full symbol sets of the APL dialects APL2, Dyalog APL, I-APL, Sharp APL, and APL.68000 are supported. All APL symbols are constructed with the symbols of the standard L^AT_EX font family. No additional fonts are needed. Standard L^AT_EX commands can be used to change the size and type style of APL symbols. Automatic conversion of APL objects is achieved by a preprocessor written in APL. L^AT_EX, `apl.sty` and the preprocessor bundled together provide an integrated high-quality APL publishing system.

1 Motivation

Why do you insist on using a notation which is a nightmare for typist and compositor and impossible to implement with punching and printing equipment currently available?

— R. A. Brooker, 1963 [Iverson 63]

Since its introduction in the early sixties, APL has been known and even become famous not so much for the power and elegance of its concepts but—to a much greater extent—for the “strange” symbols it uses. A discussion about the usefulness and difficulties of APL has remained academic for large parts of the computer science community. Many programmers never managed to get their dumb ASCII-terminals to produce the non-ASCII symbols required by the language.

With the advent of bit-mapped displays, downloadable fonts and the spread of graphical interfaces such as the X Window System the situation has changed. Specialized hardware is no longer a prerequisite for APL programming. Although it often requires some effort of customization and configuration, it is possible nowadays to turn existing hardware into an APL environment.

Troubles show up as soon as you start publishing results produced in your APL environment. (Just have a look at some books on APL, where the APL passages had to be pasted in!) Since