Macros

That ol' devil `\expandafter`

Don Hosek

Every TEX hacker knows that `\expandafter` is an essential part of one's macro vocabulary. Since TEX is a macro-based language, controlling the order of macro expansion is essential to making certain effects work.

However, because in practice `\expandafter` commands tend to come in swarms, it is often difficult to follow all but the most common idioms when looking at TEX macro code.

I recently was faced with this problem in the development of the qstyle core macros which I use in developing style and class files. The truth of the matter is that there is *never* code written which is not modified later. I consider it an essential matter to document every bit of code I write, when I write it, or I'll never be able to maintain it.

The qstyle code contains some relatively simple idioms like

```
\expandafter\expandafter\expandafter
   \A
\expandafter
   \B
   \C
```

which would cause the order of expansion of `\A`, `\B` and `\C` to be reversed.[1]

[1] Highly recommended reading on this topic would include Stephen Bechtolsheim's article in *TUGboat* Vol. 9, No. 1, which appears in a modified form in his book, *TEX in Practice*.

But for non-standard expansions, say, to expand \B, then \C, then \A,[2] the code may become a little more convoluted to say the least. In this case, an unambiguous representation of the order of expansion comes in handy. Towards this end, I came up with an indentation-based approach to getting the job done. The basic principle is to have the first level of indentation indicate one set of macros expanded, and keep indenting to complete the job.The first macro I used this on was the \QNameLetName macro which allows me to take the names of two control sequences and \let the equivalent \csname of the first to the \csname of the second. The macro is much simpler if the first \csname is expanded before the second since we then have only one token to skip over.[3] In traditional indentation schemes, the code would be hard to follow, but using an indentation based on expansion order, it becomes much easier to follow.

```
\def\QNameLetName#1#2{%
   \expandafter
      \expandafter
   \expandafter
         \let
   \expandafter
      \expandafter
   \csname#1\endcsname
      \csname#2\endcsname
}
```

Annex

During the review process, it was pointed out by Victor Eijkhout that there is a simpler solution to the programming problem above:

```
...
\expandafter\let
  \csname #1\expandafter\endcsname
     \csname #2\endcsname
...
```

Note that while this is a simpler and more efficient solution, the argument above about intelligent indentation stands.

◇ Don Hosek
Quixote Digital Typography
555 Guilford
Claremont, CA 91711
dhosek@quixote.com

[2] By the way, this can only be done if \B expands to a single token.

[3] One of the joys of TeX: Sometimes when we expand something, we end up with *fewer* tokens than when we started.