

Bernd Raichle

Check if defined, no side effects

LaTeX and other macro packages include a test if a control sequence is already defined using `\csname...\endcsname`. In principle the following definition is used:

```
\def\@ifundefined#1#2#3{%
  \expandafter\ifx\csname #1\endcsname\relax#2\else#3\fi}
```

The use of `\csname...\endcsname` has the side effect that an undefined control sequence will be defined as `\relax` and a simple test using `\ifx<control sequence>\undefined` will fail.

To avoid this side effect, you can change the definition above introducing a group “around” `\csname`:

```
\def\@ifundefined#1#2#3{%
  \begingroup \expandafter\expandafter \endgroup
  \expandafter\ifx\csname #1\endcsname\relax#2\else#3\fi}
```

Note: This definition is not fully expandable, thus it will fail in expansion-only contexts where the original definition will work.

If you are using ϵ -TeX, the new `\ifcsname...\endcsname` primitive can be used instead.

During the presentation of this pearl at the BachoTeX 2006 pearls session, two alternatives to this trick has been proposed. Worthy to note, that the macro above use more `\expandafter`'s then actually needed. Once we say

```
\def\@ifundefined#1#2#3{%
  \begingroup \expandafter
  \ifx\csname#1\endcsname\relax\endgroup#2\else\endgroup#3\fi}
```

we get exactly the same effect in a bit more efficient way. But there is one more sticky problem in all the constructions above; every control sequence defined as `\relax` (and `\relax` itself) is treated as undefined. The solution seems to be

```
\def\@ifundefined#1#2#3{%
  \begingroup \expandafter \endgroup \expandafter
  \ifx\csname#1\endcsname\undefined#2\else#3\fi}
```

Here we expand `\csname...\endcsname` before `\endgroup`, but the condition is fixed outside the group. Thus, `\undefined` instead of `\relax` can be used for comparison.