1 Introduction

The 2015 release of Xe\TeX\ introduced a new command \texttt{\Ucharcat}, this is an extension of the \texttt{\Uchar} command that has been available in \TeX\ and \luatex\ for some time. It takes a second integer value, that specifies the category code of the token to be produced. This allows character tokens to be constructed via \textit{expansion}, which has many potential uses in producing expandable case changing, numeric counter representations, etc.

\texttt{\Uchar 65 12} produces a catcode 12 \texttt{A} for example.

This package provides a lua implementation of \texttt{\Ucharcat} for use with \luatex, it silently accepts \TeX\ and does nothing in that case if \texttt{\Ucharcat} is defined.

The main difference between the lua implementation and the \TeX\ primitive is that the lua implementation takes \textit{two} expansions to produce the token.

\texttt{\edef\tmp{\Uchar 65 11}}

is the same as \texttt{\def\tmp{A}} with both systems but

\texttt{\expandafter\def\expandafter\tmp\expandafter{\Uchar 65 11}}

the same as

\texttt{\def\tmp{A}}

with \TeX, but in \luatex\ it is equivalent to

\texttt{\def\tmp{\directlua{\UcharcatLua() 65 11}}}

2 Examples

This section will be omitted if this document is not processed with a suitable format.

\footnote{This file has version number v0.03, last revised 2015/11/19. Please report any issues at https://github.com/davidcarlisle/dpctex/issues}
• \texttt{\textbackslash Ucharcat 65 11}
  A is a capital A.

• \texttt{\textbackslash Ucharcat 65 12}
  This is a catcode 12 A: yes.

• \texttt{\textbackslash Ucharcat 65 1} and \texttt{\textbackslash Ucharcat 65 2}
  \textbf{Bold is grouped} by catcode 1 and 2 A.

3 Implementation

Note that the current implementation uses \texttt{\textbackslash directlua} and a dedicated \texttt{luatex} catcode array. Hans hagen made some useful comments and pointers to alternative implementation using \texttt{\textbackslash luafunction} in the \texttt{luatex} list thread \url{http://tug.org/pipermail/luatex/2015-May/005199.html}. For now keeping with the simpler initial approach as there is no built in support for \texttt{\textbackslash luafunction} in the \LaTeX\ format yet. (No hook to save function definition in the format)

1 (*package*)

On classic \TeX or old \XeTeX, stop.

2 \texttt{\ifx\directlua\@undefined}
3 \texttt{\ifx\Ucharcat\@undefined}
4 \texttt{\ifx\XeTeXinterchartokenstate\@undefined}
5 \texttt{\PackageError{ucharcat}}
6 \texttt{\{string\Ucharcat\space may only be used with xetex and luatex\}}
7 \texttt{\{skipping package\}}
8 \texttt{\else}
9 \texttt{\PackageError{ucharcat}}
10 \texttt{\{string\Ucharcat\space is defined in xetex releases from 2015 only\}}
11 \texttt{\{skipping package\}}
12 \texttt{\fi}
13 \texttt{\fi}
14 \texttt{\expandafter\endinput}
15 \texttt{\fi}

Current \LaTeX\ formats (from 2015/11/01) define an allocator for catcode tables otherwise just use one (near) the top of the range (hex 7FFF).

16 \texttt{\ifx\newcatcodetable\@undefined}
17 \texttt{\chardef\ucharcat@table"7000}
18 \texttt{\directlua{\texttt{\textbackslash initcatcodetable}}}({"\textbackslash initcatcodetable"})
19 \texttt{\initcatcodetable\ucharcat@table}
20 \texttt{\else}
21 \texttt{\newcatcodetable\ucharcat@table}
22 \texttt{\fi}

\texttt{lua print function}
23 \texttt{\directlua{\%}}
24 \texttt{\local nt = newtoken or token}
function UcharcatLua()
    local mych = nt.scan_int()
    local mycat = nt.scan_int()
    tex.setcatcode(\the\numexpr\ucharcat@table\relax,mych,mycat)
    tex.sprint(\the\numexpr\ucharcat@table\relax,unicode.utf8.char(mych))
end
}

\TeX{} wrapper.
def\Ucharcat{\directlua{UcharcatLua()}}

//package