The ucharcat Package*

David Carlisle

August 23, 2015

1 Introduction

The 2015 release of XeTEX introduced a new command `\Ucharcat`, this is an extension of the `\Uchar` comand that has been available in XeTEX 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 expansion*, which has many potential uses in producing expandable case changing, numeric counter represntations, etc.

`\Uchar 65 12` produces a catcode 12 `A` for example.

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

The main difference between the lua implementation and the XeTEX primitive is that the lua implementation takes *two* expansions to produce the token.

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

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

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

the same as

```
\def\tmp{A}
```

with XeTEX, but in luaTEX it is equivalent to

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

2 Examples

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

*This file has version number v0.02, last revised 2015/08/23. Please report any issues at https://github.com/davidcarlisle/dpctex/issues

- `\Ucharcat 65 11`

  A is a capital A.

- `\Ucharcat 65 12`

  This is a catcode 12 A: yes.

- `\Ucharcat 65 1` and `\Ucharcat 65 2`

  Bold is grouped by catcode 1 and 2 A.

3 Implementation

Note that the current implementation uses `\directlua` and a dedicated luatex catcode array. Hans hagen made some useful comments and pointers to alternative implementation using `\lusfunction` in the luatex list thread `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 `\luafunction` in the LaTeX format yet. (No allocator macros or hook to save function definition in the format)

```
1 ⟨*package⟩
```

On classic TeX or old XeTeX, stop.

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

If some package has defined an allocator for catcode tables use it, othewise just use one (near) the top of the range (hex 7FFF).

```
16 \ifx\newcatcodetable\@undefined
17   \chardef\ucharcat@table"7000
18   \directlua{tex.enableprimitives("",{"initcatcodetable"})}
19   \initcatcodetable\ucharcat@table
20 \else
21   \newcatcodetable\ucharcat@table
22 \fi
```

lua print function

```
\directlua{%
function UcharcatLua()
  local mych = newtoken.scan_int()
  local mycat = newtoken.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⟩