percentchar

`ucs.sty` - Unicode Support

Dominique P. G. Unruh

July 4, 2002

Contents

1 Usage

Simply use `\usepackage{ucs}` and `\usepackage[utf8]{inputenc}`, then you will be able to write your LaTeX-Documents in UTF-8.

You can access a Unicode character with `\unichar{`⟨*code*⟩`}`, even when the active input encoding is not `utf8`.

`\SetUnicodeOption` An unicode character can have an default glyph macro and several glyph macros associated with options. If one of these options is set, the associated macro is used, otherwise the default macro. If several associated options are set, an error is yielded. You may activate an option ⟨*name*⟩ by including it in the option list while loading the ucs package, or by using `\SetUnicodeOption{`⟨*name*⟩`}`. To deactivate an option, prefix its name by `no`. Note that you must load `ucs.sty` before `\usepackage[utf8]{inputenc}` if you want to supply options. Any option which you want to use must be used at least once in the preamble.

When you activate an option, you can supply a priority as optional argument. If there are several glyphs for a given code position, the one having the option with the highest priority is taken (an error is yielded in case of ambiguity). If you do not supply a priority, 100 is taken as default. “Normal” glyphs are associated with the option `default`, which is initially activated with a priority of 0.

NB: UTF-8 characters are interpreted by TeX as a sequence of commands, so don't use calls like `\macro ä` instead of `\macro{ä}` (this does not apply to ASCII characters).

This input encoding does not change the fontencoding automatically. You can use the package `autofe` for that purpose (`http://www.unruh.de/DniQ/latex/unicode/content/unsupported/autofe.sty`), but this is not yet in a release ready state and thus unsupported and undocumented.

1.1 Special options

Several options have a special hardcoded meaning:

`combine`
- `combine`: Activates combining mode. See section 1.3.

`default`
- `default`: This option contains all characters, which are not explicitly associated with another option. But see also the option `document`. This option is activated per default, but has priority 0, i.e. any other activated option is prefered, unless its priority is explicitly given to be smaller.

`document`
- `document`: Every character you define in your document using `\DeclareUnicodeCharacter` has the option `document`. This option is activated per default having the priority 1000, therefore manually declared characters take precedence over all other characters, unless some other option has explicitly gotten a higher priority.

`fasterrors`
- `fasterrors`: When used, the name of an unicode character is not included in error messages any more, which runs much faster.

`graphics`
- `graphics`: When used, unknown characters are replaces by GIFs downloaded from `unicode.org`. Commands to download and convert these are executed if -shell-escape is passed to LaTeX, otherwise they are proposed to the user in a warning message. A UNIX-machine supporting the commands `wget`, `giftopnm` and `pnmtops` is assumed.

`savemem`
- `savemem`: When used, only the character needed at the moment is loaded, not a whole page. This slows down operation, but saves space in the TeX-pool, especially with sparsly used character set like kanji. Use this, if you get an out of pool error or similar from TeX. If you change the state of this option during the run, you may get inexpected results.

`warnunknown`
- `warnunknown`: When used, an unknown unicode character does not generate an error, but a warning.

1.2 Normal options

The options described here are—strictly spoken—not part of `ucs.sty`, but are defined by the unicode data files. They are included here for convenience.

autogenerated

- `autogenerated`: This enables the characters, which are autogenerated as composition of other characters according to the informations in the UnicodeData.txt. These may or may not look good. You may have to define the `\unicodecircle`, `\unicodesquare`, `\unicodevertical`, `\unicodewide` and `\unicodesmall` macros to let all autogenerated macros work. Furthermore you may have to set some other options, when the autogenerated characters are build out of characters, which are not in the default set.

cjkbg5

- `cjkbg5`: See the explanation for `cjkjis` below and substitute C40 by C00, JIS by BIG-5 and `kanji48` by some appropriate font which has BIG-5 encoding (e.g. one of the "Arphic AR PL * Big5" fonts).

cjkhangul

- `cjkhangul`: See the explanation for `cjkjis` below and substitute C40 by C61, JIS by "KSC5601 hangul syllables" and `kanji48` by some appropriate font which has KSC5601 encoding and hangul syllables (e.g. the `han` or the `han1` font from CJK-LaTeX).

cjkjis

- `cjkjis`: This enables the use of C40 (JIS) encoded fonts. You need to have the `c40*.fd` files which are contributed with the package `CJK` and the `kanji48` font installed for this. Further you have to load the fontencoding `C40` (an option to the package `fontenc`).

  It is not necessary to load the package `CJK`. If you want to use it nevertheless take care of the following:

  - Load `CJK` before `fontenc`, or quite strange errors will occur.
  - Load `CJK` with option `encapsulated`, or it will overwrite some of the UTF8 input encoding.
  - Don't use the `CJK` environment, it destroys the input encoding. Use `ucjk` instead, which is a patched version and takes no arguments.

fullmathletters

- `fullmathletters`: This option has been removed. Replace all occurrences by `mathletters`.

mathletters

- `mathletters`: When using this option is set, some unicode code characters like greek or some hebrew letters generate the math mode glyphs. This option is disabled by default, because using math greek in a normal text does not look good. But you may set it in `\everymath` and `\everydisplay` and such enable the use of unicode characters in math mode. You can also use this to get a poor man's greek, it is however recommended to use the `cb` fonts instead.

postscript

- `postscript`: This option enables use of postscript, e.g. of postscript fonts. Some DVI-viewers may have problems with documents using this option, but most viewers can handle it correctly.

privatecsur

- `privatecsur`: This option enables use of characters in the private area according to the mapping by the ConScript Unicode Registry (`http://www.evertype.com/standards/csur/` and also `http://home.ccil.org/~cowan/csur/index.html`).

tipa

- `tipa`: This enables the use of the macros in the `tipa` package to display IPA symbols.

1.3 Combining mode

In some cases, Unicode documents contain sequences like U+0063 LATIN SMALL LETTER C U+0301 COMBINING ACUTE ACCENT(producing ć). In order to typeset them correctly, we cannot render U+0063 LATIN SMALL LETTER Cimmediately when it occurs, instead we have to wait whether some combining character is going to follow.

`combine` To enable this way of parsing in `ucs.sty`, we have to use the option `combine`. As long as this option is in effect, characters are not immediately rendered, but stored in a token register until they are output via

- resetting of the option (`\SetUnicodeOption{nocombine}`) or
- `\unicodecombine` the command `\unicodecombine`.

Since characters handled by `ucs.sty` are not immediately output while ASCII characters are handled by TEXand directly rendered, you should not mix ASCII characters and non-ASCII characters while this option is in effect. Thus to obtain the above glyph, you cannot use

```
\SetUnicodeOption{combine}có\SetUnicodeOption{nocombine}
```

(where ó is U+0301 COMBINING ACUTE ACCENT), instead you can use one of the following constructions:

- `\SetUnicodeOption{combine}\unichar{"63}ó`
  `\SetUnicodeOption{nocombine}`
- `\SetUnicodeOption{combine}\unicodevirtual{c}ó`
  `\unicodevirtual` `\SetUnicodeOption{nocombine}` (`\unicodevirtual` takes arbitrary LATEX code and inserts it, as thou it was a Unicode character; do not use Unicode characters inside `\unicodevirtual`).
- `\SetUnicodeOption{combine}\myverbatim|có|`
  `\SetUnicodeOption{nocombine}`
  where `\myverbatim`[1] is a command similar to `\verb`, but setting the catcodes of the ASCII characters to 13 (active) and then defining character no. n to expand to `\unichar{`n`}`.

In cases where you only want to render occasional words containing combining characters and *no* ASCII, you can use a macro like

```
\newcommand\combword[1]{\SetUnicodeOption{combine}#1
    \SetUnicodeOption{nocombine}}
```

and then simply typeset the concerning word as an argument to `\combword`.

1.4 Defining unicode data

A unicode character may be defined by

`\DeclareUnicodeCharacter` `\DeclareUnicodeCharacter{`⟨*code*⟩`}{`⟨*macro*⟩`}`

or, when it is to be associated with a special option, by using

`\DeclareUnicodeCharacterAsOptional`

`\DeclareUnicodeCharacterAsOptional{`⟨*code*⟩`}{`⟨*option*⟩`}{`⟨*macro*⟩`}`
where ⟨*code*⟩ is the unicode character number, ⟨*option*⟩ the associated option and ⟨*macro*⟩ the glyph's macro.

This definition is local.

In the automatically loaded data files `\uc@dclc` should be used instead.

`\DeclareUnicodeOption`

An option ⟨*option*⟩ can be defined by
`\DeclareUnicodeOption[`⟨*pkg*⟩`]{`⟨*option*⟩`}`.
If ⟨*pkg*⟩ is supplied, the option is set, if the package ⟨*pkg*⟩ is loaded.

`\LinkUnicodeOptionToPkg`

You can add further packages, which automatically set an option, by
`\LinkUnicodeOptionToPkg{`⟨*option*⟩`}{`⟨*pkg*⟩`}`.

If a character c is unknown, it is looked up in in the `uni-`n`.def`-file, whereby $n = \lfloor \frac{c}{256} \rfloor$. So characters which are not document specific, should be defined in those file. For generating them, you should you the program `makeunidef.pl`.

You may find `\dirtyunicode` and `\UnicodeNeeds` interesting for writing glyphs macros, see in the implementation section.

1.5 Known problems

Note, that if a character from some not yet loaded Unicode page appears, a file has to be loaded. If this appears inside a word, kerning and ligatures do not work at that position.

Further there are some commands, which expand their arguments in non-executing contexts, this makes it impossible for `ucs.sty` to load the character definition file at this place if this has not yet been done. In this case the concerning character is replaced by some message that you have to use `\PrerenderUnicode{`...`}`.

Both problems can be solved by preloading the offending characters. If you have for example U+03B1 GREEK SMALL LETTER ALPHA, which should be preloaded, simply use

`\PreloadUnicodePage`

- `\PreloadUnicodePage{3}` (the argument to `\PreloadUnicodePage` is $\lfloor \frac{n}{256} \rfloor$, where n is the number of the character (here 0x3B1 = 945)) or

`\PrerenderUnicode`

- `\PrerenderUnicode{`α`}` (the argument to `\PrerenderUnicode` can contain any LaTeX code, which is then rendered in an hbox, all still unknown characters a loaded and the result is thrown away; do not use e.g. `\footnote` or other commands which might not like to be executed several times).

2 Thanks

Thanks to...

- Michel Goossens who supplied many characters (e.g. Vietnamese, polytonic Greek),

[1] This command is not provided by `ucs.sty`, but some similar command may appear in future.

- Manuel Kauers for testing my package before the first upload,
- Werner Lemberg, who wrote the `CJK` package, where I got the font definitions in `cenccmn.tex` from,
- Karsten Tinnefeld for many of the glyph macros in `cyrillic.ucf`,
- Pablo Rodriguez for reporting many bugs, especially concerning the interaction between `ucs.sty` and other packages.
- Stefan Röhrich for testing my package before the first upload,
- the authors of all those many LaTeX-packages for different scripts.