ConTEXt

title : **ConTEXt User Module**
subtitle : **The Transliterator**
author : **Philipp Gesang**
date : **January 13, 2011**

```
\writestatus{loading}{Transliteration from non-Latin scripts}

\unprotect

\definenamespace [TRL] [
  name=transliterate,
  type=module,
  setup=list,
  parent=TRL,
  style=no,
  version=hg-r16,
  comment=Transliteration from non-Latin scripts.,
]

\ctxlua{environment.loadluafile ("transliterator")}
```

Use the Transliterator by adding `\usemodule[transliterator]` somewhere before `\starttext`. Adjust the Transliterator through the `\setuptransliterate` command. As a first argument it accepts a set of key-value options; at present you may configure `mode` and `hyphenate`.

At first we'll set some defaults:

```
\setuptransliterate[mode=ru_old,hyphenate=cz,debug=false]
```

Possible values for `mode` are by the time of this writing: `ru`, `ru_transcript_de`, `ru_transcript_en`, `ru_old`, `all`, `iso9_ocs`, `ocs`, `ocs_gla`, `ru_cz`, `ocs_cz`, `gr` and `gr_n`. As not all fonts, even the expensive ones, support some of the most frequent unicode signs used in ISO 9 there are fallbacks for the transliterations of the weak and hard sign: `iso9_ocs_hack`, which is essentially `iso9_ocs`, and `ru_old_jer_hack`, which is essentially `ru_old`. These two transliterate and (both upper and lower case) to the more common, but non-ISO characters ' and '' respectively. Possible values for `hyphenate` are all valid CONTEXT language code, for an overview see `http://wiki.contextgarden.net/Language_Codes`. In praxi you may want to choose either Czech (the default) or Slovak (`sk`) for most transliterations from cyrillic scripts. I've not yet made up my mind concerning Greek transliteration, any suggestions are welcome.

The following will help debugging and reviewing tables. Make sure your typescript can handle the characters, in general it's no use with Latin Modern which unfortunately provides only a restricted set of the unicode range.

The user-level command to output a single substitution table is `\showOneTranslitTab{#1}`.

```
\define[1]\showOneTranslitTab{%
  \startluacode
    environment.loadluafile ("trans_tables_iso9")
    environment.loadluafile ("trans_tables_trsc")
    environment.loadluafile ("trans_tables_scntfc")
    environment.loadluafile ("trans_tables_trsc")
    environment.loadluafile ("trans_tables_glag")
    environment.loadluafile ("trans_tables_gr")
    translit.gen_rules_en()
    translit.gen_rules_de()
    translit.show_tab(translit["\luaescapestring{#1}"])
  \stopluacode
}
```

The user-level command to output all defined tables is `\showTranslitTabs`.

```
\define\showTranslitTabs{%
  \ctxlua{translit.show_all_tabs()}%
}

\def\translitDebug#1{%
  \doif{\transliterateparameter{debug}}{yes}{%
    {\ss\inmargin{\ctxlua{translit.debug_next()}} #1}%
  }%
}
```

The user-level command `\transliterate[#1]{#2}` does the job of switching to a given language (for hyphenation) and adjusting the substitution method locally. It takes an optional list `[#1]` of key-value arguments to allow ad-hoc specification of either two that deviate from the defaults set initially by means of `\setuptransliterate`.

Internally, `\dotransliterate` is called according to the CONTEXT coding style and in case the user provides `hyphenate=` or `mode=` those will be used instead of the globals. Note that this leaves the latter unchanged. Thus, in order to permanently switch to another transliteration style the user would have to set it by calling `\setuptransliterate` again.

```
\def\dotransliterate[#1]#2{%
  \bgroup\iffirstargument
    \setuptransliterate[#1]%
  \fi
    \language[\transliterateparameter{hyphenate}]%
    \ctxlua{translit.transliterate("\transliterateparameter{mode}","\luaescapestring{#2}")}%
  \egroup%
}

\unexpanded\def\transliterate{\dosingleempty\dotransliterate}

\unexpanded\def\starttransliterate{%
  \bgroup%
  \dosingleempty\dostarttransliterate%
}
```

```
\let\stoptransliterate\relax

\def\dostarttransliterate[#1]#2\stoptransliterate{%
  \iffirstargument
    \setuptransliterate[#1]%
  \fi
  \language[\transliterateparameter{hyphenate}]%
  \ctxlua{translit.transliterate("\transliterateparameter{mode}","\luaescapestring{#2}")}%
  \egroup%
}

\protect \endinput
```